Adaptive Equalizer Functions for QAM in Digital Microwave Radio Systems

Masters Thesis in Control & Communication
Linköping University, Sweden
by

MIKAEL TAPIO

Reg no: LiTH-ISY-EX-3107
Linköping, September 5, 2000

INSTITUTE OF TECHNOLOGY
LINKÖPINGS UNIVERSITET

Department of Electrical Engineering
Linköping University
S-581 83 Linköping, Sweden

Linköpings Tekniska Högskola
Institutionen för Systemteknik
581 83 Linköping

Abstract

The desired increase in data transmission rates requires more spectral efficient methods for modulation of the transmitted radio signal, e.g. 16-QAM. Therefore the radio system becomes more sensitive to corruption of the transmitted signal, which is caused by channel imperfections, e.g. multipath propagation (which introduce inter-symbol interference) and noise. To deal with these problems, equalizers are introduced on the receiving end. The choice of an adaptive equalizer becomes relevant since the channel is time-varying and the equalizer needs to "track" channel variations.

This thesis presents the channel and equalization problem, channel model, and adaptive functions for various adaptive equalizers using both trained and blind algorithms. The performance of the presented equalizers are also compared and evaluated. The focus is set on low-complexity equalizers for implementation in digital microwave radio systems.

All of the proposed equalization techniques manage to equalize a noisy and fading digital radio channel for 16-QAM signaling. The final choice of equalization method for real implementation is left together with other synchronization issues, e.g. clock synchronization and carrier recovery, for further investigation.

Keywords: Digital Microwave Radio, 16-QAM, Fading, Inter-Symbol Interference (ISI), Channel Model, Adaptive Equalizer, Linear Equalizer, Decision Feedback Equalizer (DFE), Trained LMS, Blind Equalization, CMA, FS-CMA.

Acknowledgements

I would like to thank the Digital Signal Processing unit of the SML division, Ericsson Microwave Systems AB, Mölndal, Sweden, and its manager Claes-Göran Sundberg for giving me the opportunity to study the world of digital microwave radio and equalization. Thanks also to my supervisor Christian Takvam at Ericsson for tips and hints regarding the Ericsson workplace and for guiding me to the right people. I would also like to thank Dan Weinholt and Björn Gävert of Ericsson for useful discussions regarding channel models, and thanks to my examiner Professor Fredrik Gustafsson of the Electrical Engineering Department at Linköping University for helpful comments and follow-ups.

Finally, I would like to send a big kiss and hug to my fiance Michelle who gave up the beautiful climate of Australia just to put up with me, the cold of Sweden, late nights, and if that was not enough, also weird talk about something called signal processing and for proof reading this "uninteresting and exceptionally boring" thesis. You are the best!

This report was prepared in LaTeX 2_ε.

Mikael Tapio, September 2000

To Michelle.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This is the introductory chapter and introduces the thesis by presenting the background, a short problem description, limitations, thesis outline and notations.

1.1 Background

This thesis was completed as a part of the Masters program and is the final examination for the Masters degree in Applied Physics and Electrical Engineering at Linköping University, Sweden. This work was completed in conjunction with the Microwave Communication's Digital Signal Processing unit, which is part of the Ericsson Microwave Systems company, Mölndal, Sweden.

1.2 Short Problem Description

Due to the rapid growth in wireless transmission of speech, data etc, there exists a rising demand for more spectral efficient radio communication links. Therefore we are required to look at higher level modulation schemes such as for example 16-QAM[1], which is the modulation scheme that Ericsson intend to study for future implementation in their digital microwave radio links. However, with increasing number of modulation levels and limitations on transmitted power, leading to "tighter" symbol constellations, the receiver becomes much more sensitive to signal distortion caused by the *fading* channel and impaired *noise*. Thus, the need for an *adaptive equalizer* that continuously adapts to the time-varying channel and fully recovers the distorted signal becomes evident. The purpose of this thesis is to present

[1] Explained in Section 2.2

and compare various realizable adaptive equalizer functions. The results are of special interest for Ericsson in their development of more spectral efficient digital microwave radio-links. This report aims also to function as a tutorial for people without any background in the channel/equalization problem, and therefore the style of this report can be found a bit talkative and textbook-like.

1.3 Limitations

- The resulting adaptive equalizer is required to have a low computational complexity and at low cost be simple to implement in an ASIC (application specific integrated circuit) or FPGA (field programmable gate array). Therefore the focus is set on low-complex LMS[2]-like equalizer functions.
- Independent, uniformly distributed, and zero-mean 16-QAM source at a symbol-rate of 25 MHz, i.e. 100 Mbit/s digital radio, and additive white Gaussian noise.
- This thesis deals solely with equalization and therefore assumes that we have a well-functioning carrier recovery scheme. We allow, though, for the baseband channel to introduce a phase offset which we have to compensate for in our equalization schemes.
- The work is to be carried out in baseband using the MATLAB® workspace as a tool.

1.4 Thesis Outline

Chapter 1 introduces the thesis by presenting the background, a short problem description, limitations, thesis outline and notations.

Chapter 2 discusses the basics of digital microwave radio systems with aspect to modulation schemes, channel effects and typical receiver problems.

Chapter 3 introduces the ideas behind the transmitter and receiver filters for the basic digital QAM radio system, and a model of the multipath fading microwave channel. The chapter concludes with the introduction of a complex valued discrete-time baseband channel model which includes transmitter and receiver filters.

Chapter 4 introduces the linear equalizer and various methods for updating the equalizer parameters. First we look at the structure of the

[2]Least mean square

complex linear equalizer and then we move on with various algorithms for equalizer updates such as; the zero-forcing and least-mean-square equalizer that imply a training sequence, the blind constant modulus algorithm and finally the blind decision-directed equalizer.

Chapter 5 presents the theory behind the adaptive non-linear decision feedback equalizer (DFE) together with algorithms that imply both training signals and blind updates.

Chapter 6 introduces some performance quantities, simulates and compares the various and previously introduced equalizers abilities to equalize fading and noisy microwave radio channels.

Chapter 7 presents conclusions regarding the results of the simulations and proposed equalizer techniques. It also outlines future work and recommendations.

1.5 Notations

Continuous-time signals are usually denoted by $x(t)$ and their sampled time-discrete equivalents are denoted by $x_n = x[n] = x(nT)$, where $T = \frac{1}{f_s}$ is the sampling period (f_s is the sampling frequency).

Furthermore, to simplify the notation when convolving signals and systems in time-domain, the time delay operator q is used.

When convolving a filter (or system) with a discrete signal, say $h_n = h[n]$ and $x_n = x[n]$, $n \geq 0$ (zero elsewhere) respectively, the resulting output is given by:

$$y_n = h(q)x_n = h_n * x_n = \sum_{k=0}^{\infty} h_k x_{n-k} \tag{1.1}$$

where $h(q) = \sum_{k=0}^{\infty} h_k q^{-k}$, and the time delay operator q has the function: $q^{-\ell}x[n] = x[n-\ell]$. In other words, the filter is (in *time-domain*) expressed in a polynomial $h(q)$, and the convolution of two cascaded (linear) filters is given by the product of two polynomials in q. In this thesis, the operator $h(q)$ is sometimes referred to as the filter's (or system's) impulse response.

The following tables present the abbreviations and mathematical notations used throughout the thesis.

Abbreviation	Definition
ACGC	Automatic complex gain control
AGC	Automatic gain control
ASIC	Application specific integrated circuit
BER	Bit error rate
BPSK	Binary phase-shift keying
BS	Baud-spaced
BSE	Baud-spaced equalizer
CM	Constant modulus
CMA	Constant modulus algorithm
CPD	Correct past decisions
DD	Decision-directed
DFE	Decision feedback equalizer
FIR	Finite impulse response
FPGA	Field programmable gate array
FS	Fractionally-spaced
FSE	Fractionally-spaced equalizer
i.i.d.	Independent and identically distributed
I-channel	In-phase channel
IIR	Infinite impulse response
ISI	Inter-symbol interference
LE	Linear equalizer
LMS	Least mean square
LOS	Line-of-sight
MSE	Mean square error
PAM	Pulse amplitude modulation
PLL	Phase-locked-loop
PSK	Phase-shift keying
QAM	Quadrature amplitude modulation
Q-channel	Quadrature channel
SER	Symbol error rate
SGD	Stochastic gradient descent
SNR	Signal-to-noise ratio
SQRC	Square-root-raised-cosine
ZF	Zero-forcing
ZFE	Zero-forcing equalizer

Table 1.1: Acronyms and abbreviations

Symbol	Definition
$(\cdot)^T$	Transposition
$(\cdot)^\star$	Complex conjugation
$(\cdot)^H$	Hermitian transpose (complex conjugate transpose)
$*$	Convolution operator
$\mathrm{E}\{\cdot\}$	Expectation operator
$\mathcal{L}\{\cdot\}$	Laplace transform operator
$\mathcal{F}\{\cdot\}$	Fourier transform operator
$\mathcal{F}^{-1}\{\cdot\}$	Inverse Fourier transform operator
$\log_e(\cdot)$	Complex logarithm operator
$\log_n(\cdot)$	n-logarithm operator
$\Re\{\cdot\}$	Real-part operator
$\Im\{\cdot\}$	Imaginary-part operator
$\mathrm{sgn}(\cdot)$	Sign-operator
$\nabla_{\mathbf{f}}$	Gradient with respect to $\mathbf{f}$: $\nabla_{\mathbf{f}} = \frac{\partial}{\partial \mathbf{f}_\mathrm{r}} + j\frac{\partial}{\partial \mathbf{f}_\mathrm{i}}$ where $\mathbf{f}_\mathrm{r} = \Re\{\mathbf{f}\}$ and $\mathbf{f}_\mathrm{i} = \Im\{\mathbf{f}\}$
$q^{-\ell}$	Baud-spaced delay operator: $q^{-\ell}x(nT) = x((n-\ell)T)$
$\tilde{q}^{-\ell}$	Two-times fractionally-spaced delay operator: $\tilde{q}^{-\ell}x(n\frac{T}{2}) = x((n-\ell)\frac{T}{2})$
$\delta[\cdot]$	Kronecker's delta function
$\delta(\cdot)$	Dirac's generalized delta function

Table 1.2: Mathematical notation and functions

Symbol	Definition
T	Symbol period
$s(t)$, s_n	Source symbol process
$h(t)$, $H(f)$	Rummler's fading channel model
$\mathrm{B_{dB}}$	Notch depth in dB
Δf	Notch displacement from center frequency
$\hat{s}_n$	Hard (quantized) source symbol estimate
δ	Overall system delay or decision delay
$v(t)$, v_n	Additive noise process
L_c	FIR channel length
L_f	FIR linear equalizer or DFE feedforward filter length
L_g	FIR DFE feedback filter length
$c(q)$	Baseband channel filter (including transmitter and receiver filters): $c(q) = c_0 + c_1 q^{-1} + \ldots + c_{L_c-1} q^{-L_c+1}$
$f(q)$	Linear equalizer filter or DFE feedforward filter: $f(q) = f_0 + f_1 q^{-1} + \ldots + f_{L_f-1} q^{-L_f+1}$
$g(q)$	DFE feedback filter: $g(q) = g_1 q^{-1} + \ldots + g_{L_g} q^{-L_g}$
$h^{\mathrm{tot}}(q)$	Equalized impulse response (channel-equalizer combination)
$\mathbf{f}$	Linear equalizer filter or DFE feedforward filter tap-weight vector
$\mathbf{g}$	DFE feedback filter tap-weight vector
$\boldsymbol{\Theta}$	Total DFE tap-weight vector (including both feedforward and feedback filter taps)
r_n	Received samples (baseband channel output)
$\mathbf{r}_n$	Regressor vector of received samples for linear equalization
$\mathbf{s}_{n-\delta}$	Regressor vector of decision delayed training symbols for decision feedback equalization
$\hat{\mathbf{s}}_n$	Regressor vector of hard symbol estimates for decision feedback equalization
$\boldsymbol{\Phi}_n$	Total DFE regressor vector (including both received samples and training symbols)
$\hat{\boldsymbol{\Phi}}_n$	Total DFE regressor vector (including both received samples and hard symbol estimates)

Table 1.3: System quantities

Symbol	Definition
y_n	Output of linear equalizer (soft symbol estimate)
$\tilde{y}_n$	ACGC output for linear equalization
z_n	Output of decision feedback equalizer (soft symbol estimate)
$\tilde{z}_n$	ACGC output for decision feedback equalization
ε_n	Error function or error signal
G_n	Recovered gain in the ACGC algorithms
ϕ_n	Recovered phase in the ACGC algorithms
μ	Adaptation gain (step-size) in equalizer tap-update algorithms
λ	Adaptation gain (step-size) in the ACGC's gain control algorithm
α	Adaptation gain (step-size) in the ACGC's phase control algorithm
σ_s^2	Variance of source process
σ_v^2	Variance of noise process
κ_s	Normalized source kurtosis
κ_g	Normalized kurtosis of a Gaussian process
γ	Source dispersion constant

Table 1.4: System quantities continued

Chapter 2

Digital Microwave Radio Systems

This chapter discusses the basics of digital microwave radio systems with aspect on modulation schemes, channel effects and typical receiver problems.

2.1 Overview

Communication by technical means is so common in our daily lives today that we do not consider the large complexity of the systems involved. We have wireless systems for transmitting data, speech etc, and devices such as mobile phones have also become something that is available for most people of modern society.

The basic purpose of a transmitter/receiver system is to transmit a message from an information source to a user of the information. Modern communication systems are very complex with thousands of system components. To get a general view we look at the system from a high abstraction level. The communication system consists, along with the information source and the user, of three basic elements; transmitter, channel, and receiver.

The transmitter transforms the message into a form that is suitable for transmission over the specific channel. The transmitted signal is centered at a frequency much higher than the highest frequency component of the message signal, i.e. modulated onto a carrier and transmitted over the channel. Due to channel imperfections, the channel will distort the transmitted signal and typically both interfering signals and noise are added (to the original signal). The receiver receives a corrupted version of the transmitted signal and its objective is therefore to reconstruct the original message as accurately as possible and deliver an estimate to the user.

Transmitted power and channel bandwidth are the primary resources a radio system designer has to use as efficiently as possible. The usage of power and bandwidth is regulated and it is needless to say that there is a lot of competition about the available frequency bands.

Today, modern digital techniques dominate the development of communication systems. The ever-increasing development of digital components has made it possible to design more complex radio systems with better performance.

2.2 Digital Radio Basics

The digital radio system consists of several components: source encoder, channel encoder, modulator, channel, demodulator, channel decoder, and source decoder. These components will not be explained in detail as they are described in [1]. Basically, the source encoder makes a compact discrete representation of the source information, and the channel encoder adds redundancy to the data stream, which is used in the receiver to decode the data stream in a reliable way. Furthermore, the modulator maps blocks of the data stream onto a corresponding symbol resulting in a symbol stream which is then passed onto a *pulse shaping* filter as in Figure 2.1 to form analog symbols. The sequence of analog symbols forms the output signal and is transposed to a frequency band which is suitable for transmission over the channel. There are different alphabets that the symbols can be drawn from and the more symbols the alphabet contains without increasing the average transmitted power, the more spectral efficient the communication system becomes. Two common modulation schemes are *M-ary phase-shift keying* (M-PSK) and *M-ary quadrature amplitude modulation* (M-QAM). In these schemes the symbols are drawn from a length-M alphabet of complex symbols. The symbol constellations for 8-PSK and 16-QAM are shown in Figure 2.2 where each constellation point corresponds to $\log_2(M)$ binary data bits, i.e. three and four bits for 8-PSK and 16-QAM, respectively.

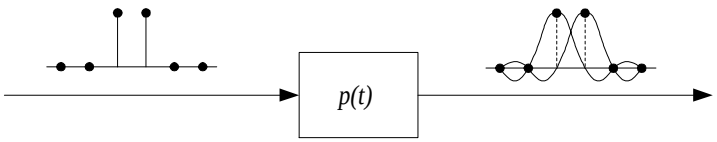

Figure 2.1: Pulse shaping using a Nyquist pulse shaping filter $p(t)$

On the receiving end, the components have the inverse function of those on the transmitting end. The demodulator converts the received waveform to an estimate of the transmitted symbol, the channel decoder reconstructs the data stream by using the redundancy in the transmitted data stream,

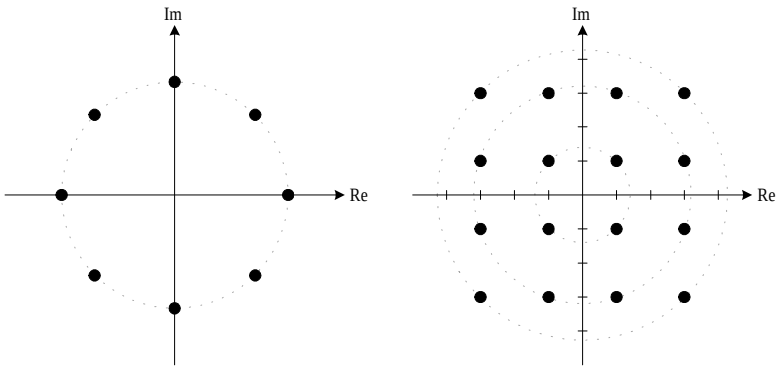

Figure 2.2: The 8-PSK (left) and the 16-QAM (right) constellations

and the source decoder reconstructs the original signal with some errors which are introduced by the communication system.

Figure 2.3 shows how a complex valued QAM symbol sequence is transmitted over a channel, received and sampled [11].

2.3 Channel Effects

2.3.1 Fading

In radio communication systems the transmitted signal is distorted by the channel. There are different forms of distortion that the channel can introduce such as, for example, signal attenuation (fading), additive noise etc. One major channel imperfection in our system is *frequency selective fading* where the channel attenuates the signal at certain frequencies creating deep *notches* in the signal spectrum. *Flat fading*, in comparison to frequency selective fading, is when the signal is constantly attenuated over the whole receiver bandwidth. Fading is introduced because the signal can propagate through the channel using different paths (as shown in Figure 2.4), i.e. *multipath propagation*. The received signal is therefore a sum of several signal rays which interfere constructively or destructively with each other at the receiver, creating unwanted fades in the received signal spectrum. Figure 2.5 shows how the spectrum of a baseband signal consisting of real valued symbols is shifted up to radio frequency (RF) and then distorted by the channel's spectrum, resulting in a down-converted distorted baseband spectrum. As we know from the complex analysis, a signal has a spectrum that is symmetric around the origin if and only if the signal is real valued, and although we transmitted a real valued signal we received a complex valued one. This implies that the channels transfer function is complex valued. The real part of this transfer function is called the *in-phase-* or *I-channel*,

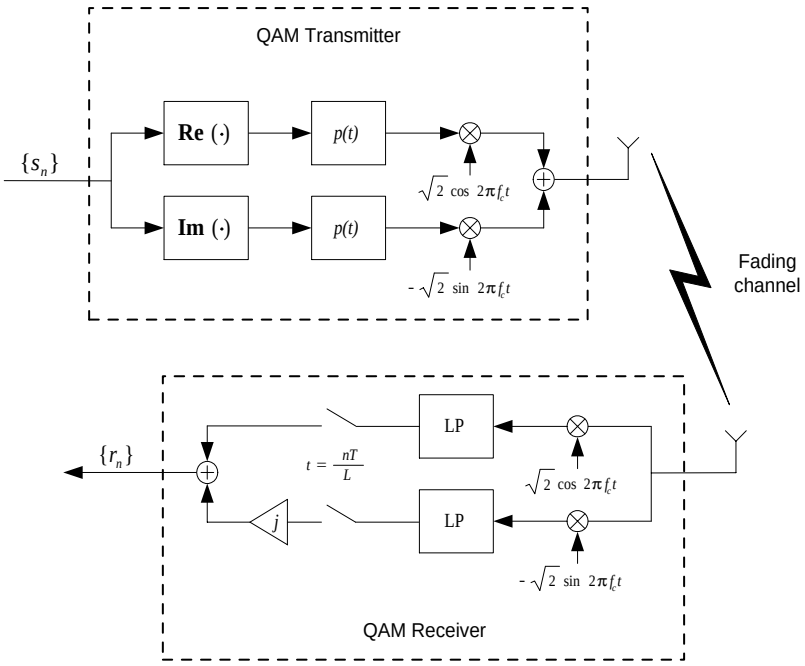

Figure 2.3: Transmission and reception of discrete complex symbols

whereas the imaginary part is called the *quadrature-* or *Q-channel*.

2.3.2 Inter-Symbol Interference

The fact that the received spectrum is not flat means that the received signal is not white, implying that the received symbols are correlated. In the time-domain this means that a single transmitted pulse will be received as a pulse train and therefore transmitted pulses will interfere with each other and thereby cause *inter-symbol interference* (ISI). The channel is said to be (*time-*) *dispersive*.

A typical channel impulse response is shown in Figure 2.6 which displays the real part of a complex valued channel impulse response. If the length or *delay spread* of the response is much longer than the symbol period then adjacent symbols will interfere with each other and cause ISI.

The effect of ISI when detecting a transmitted symbol is best illustrated by convolving the transmitted symbol sequence, $\{s_n\}$, with the discrete-time channels response, $\{c_n\}$, to a single transmitted pulse to get the received

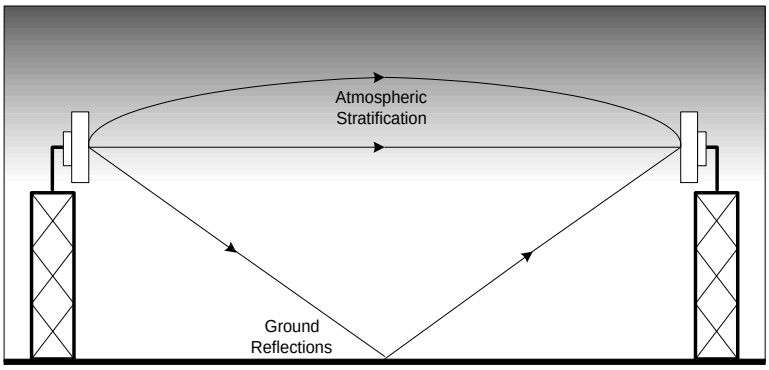

Figure 2.4: Multipath propagation

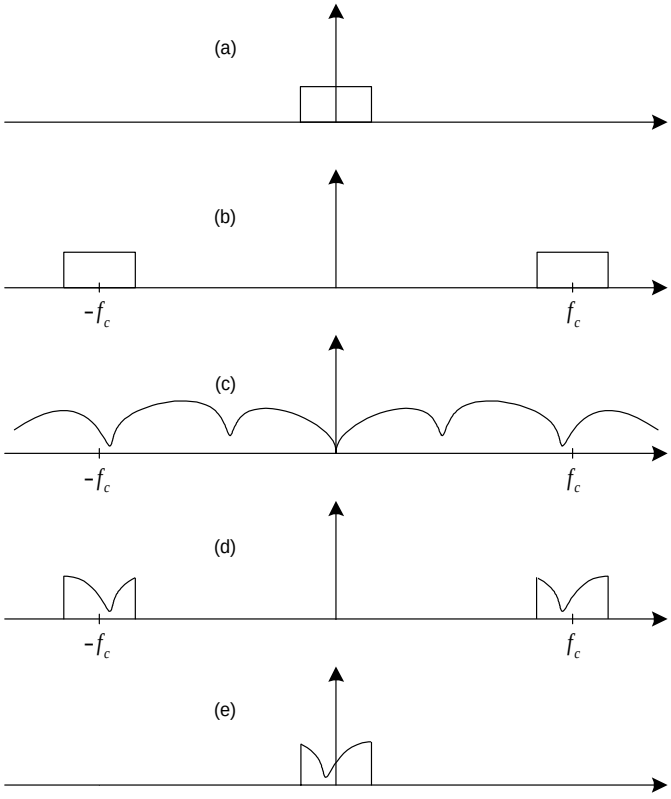

Figure 2.5: (a) Baseband signal spectrum, (b) transmitted up-converted spectrum, (c) fading channel spectrum, (d) received spectrum, and (e) received baseband spectrum

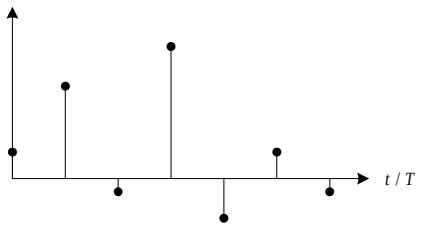

Figure 2.6: Example of I-channel pulse response that leads to ISI

sequence, $\{r_n\}$:

$$r_n = \sum_k c_k s_{n-k} + v_n = c_\delta s_{n-\delta} + \underbrace{\sum_{k \neq \delta} c_k s_{n-k}}_{\text{ISI}} + v_n, \quad n = 0, \pm 1, \ldots \tag{2.1}$$

where v_n are additive noise samples. It can easily be seen that if all the samples in the channel response are equal to zero, except for $c_\delta \neq 0$, then $r_n = s_{n-\delta}$, for some integer delay δ. But if this is not the case we then have ISI. The only channel pulse response that does not give any ISI is the one that is in the form of a (delayed) Kronecker delta function:

$$c_k = \delta[k - \delta] = \begin{cases} 1 & \text{for } k = \delta \\ 0 & \text{elsewhere} \end{cases} \tag{2.2}$$

This is a slight modification of the truth since the channels pulse response is surely not equal to unity for $k = \delta$, and more likely equal to some other complex scalar. This gives us that the received symbol is attenuated and phase-shifted, but still *not* corrupted by ISI, which is our focus here. This former problem is usually compensated for by using an *automatic gain control* (AGC) device and some *carrier recovery* scheme, i.e. *phase* (and frequency) *offset* recovery, which together can compensate for unwanted attenuation and turn the received constellation back into its original place.

2.3.3 Noise

The receiver will also experience impaired noise which is added to the received signal. This impaired noise is often in the nature of *thermal noise*. Thermal noise is usually modeled as a white, Gaussian process, and is assumed to be uncorrelated with the received signal. We have to be careful when designing a receiver so we do not amplify the noise too much, which would create severe difficulties in the symbol detection.

A common measurement of the level of noise impairment is the *signal-to-noise ratio* (SNR) which is defined as:

$$\mathrm{SNR} = \frac{\mathrm{E}\{|x(t)|^2\}}{\mathrm{E}\{|v(t)|^2\}} \tag{2.3}$$

or in decibel:

$$\mathrm{SNR}_{\mathrm{dB}} = 10\log_{10}(\frac{\mathrm{E}\{|x(t)|^2\}}{\mathrm{E}\{|v(t)|^2\}}) \tag{2.4}$$

where $x(t)$ is the received signal without noise and $v(t)$ the noise process. As a future reference for this thesis, SNRs in the range of $10-25$ dB are considered relatively low, and in the range $25-50$ dB mediate, and everything above 50 dB is considered relatively high.

2.4 Other Problems

We will also encounter some unavoidable additional problems in digital radio communications such as carrier recovery and synchronization issues. A few of these problems are mentioned here briefly.

2.4.1 Carrier Recovery

Typical problems we always have to struggle with are the ones that arise when we down-convert from radio frequency to baseband. Due to imperfections in the *local oscillators* we do not have exactly the same carrier frequency at the receiver as we have at the transmitter, which gives us an unknown *carrier frequency offset* and this in turn brings the received constellation to rotate. We also have an unknown *carrier-phase offset* between the transmitters modulation and the receivers demodulation that makes the receiver *non-coherent*, and the imperfections in the local oscillators will also cause some *phase noise* (or *phase jitter*), i.e. the constellation rotates back and forth. There are methods of estimating and then compensating for carrier frequency and phase offsets. These so called methods are by one name called *carrier recovery*.

Figure 2.7 summarizes the effects on a received 16-QAM signal constellation that are caused by noise, reflections (ISI), phase noise, and frequency offset.

2.4.2 Clock Synchronization

Another important issue is *clock synchronization* or *timing*, which is associated with the timing when sampling the received signal. With "bad" timing, we can get ISI in a non-fading situation just by sampling at the wrong time

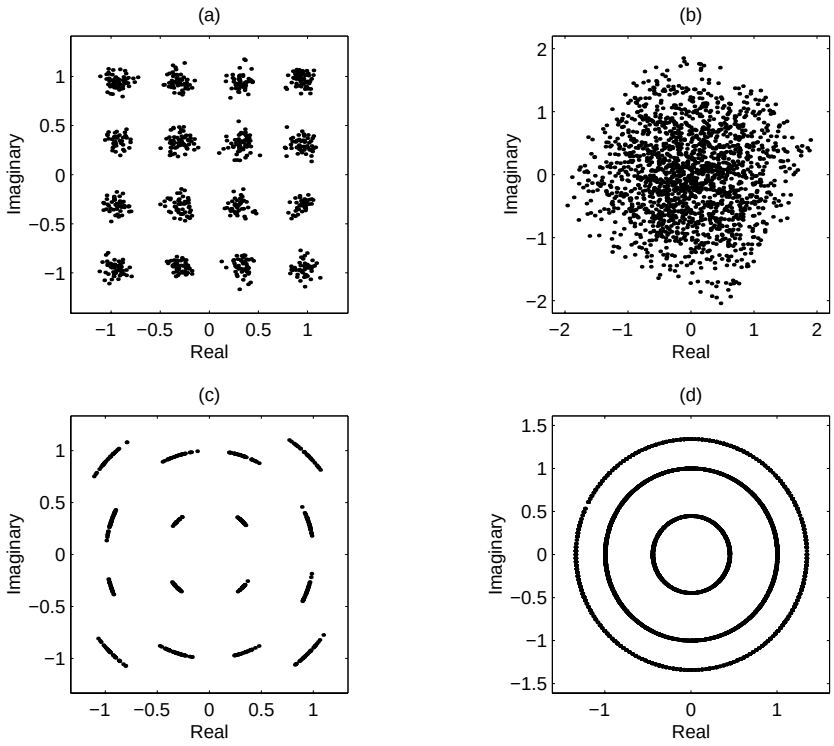

Figure 2.7: (a) Additive noise, (b) strong reflection (ISI), (c) phase noise, (d) frequency offset

instances. Figure 2.8 shows how a timing offset can lead to a discrete-time channel response that will lead to ISI. There are special methods for clock recovery, but they are not discussed in this thesis.

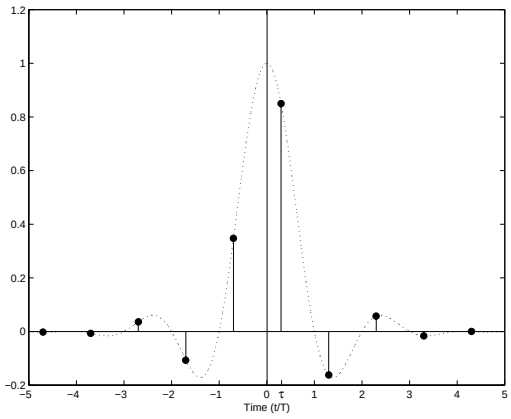

Figure 2.8: Example of a timing offset τ that will lead to ISI. Instead, it would be optimal to sample at $t = nT,\ n = 0, \pm 1, \pm 2, \ldots,$ as this would not give ISI

2.5 Equalization

In the previous sections the communication problems were introduced. As it can be seen in Figure 2.7, the ISI together with noise make it impossible to detect the transmitted symbols without some kind of pre-equalization. The equalizers task is to eliminate ISI without enhancing the noise. In frequency domain, the equalizer "flattens" the received spectrum, therefore given the name *equalizer*. There are various equalizers based on transmission of training sequences and blind equalizers that operate without any pre-training. These will be discussed more in detail in the later chapters. Figure 2.9 shows the effect of the equalizer and how a transmitted source symbol sequence that is distorted by noise and ISI is passed onto an equalizer which reconstructs the symbols and then passes them onto a *decision device*[1] (or *slicer*) which maps the equalized symbols onto a 16-QAM constellation. If the equalizer works well, the quantized symbols are equal to the transmitted source symbols.

[1] Presented in Section 4.3.4.

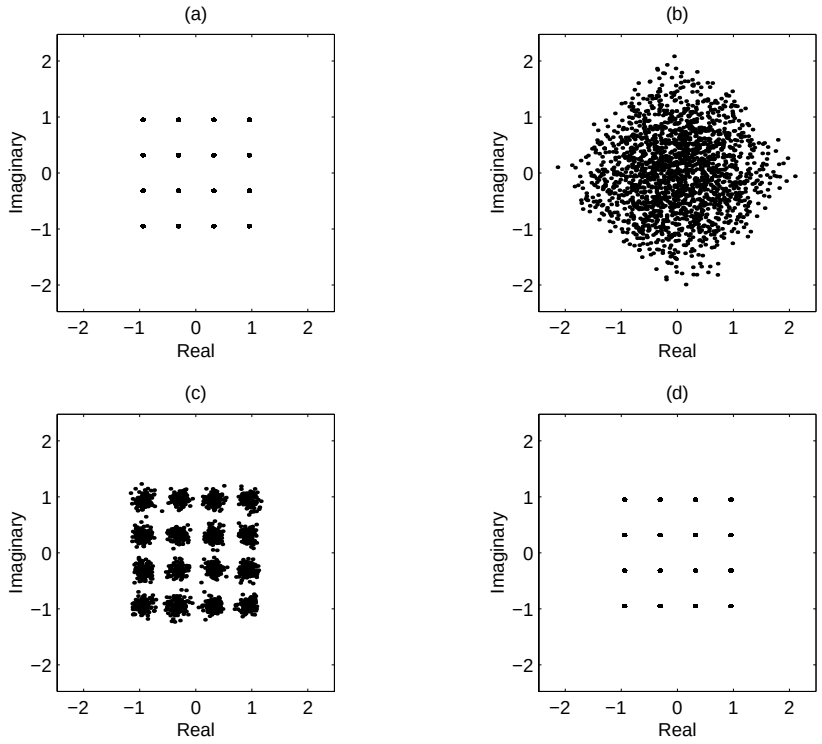

Figure 2.9: (a) Transmitted source constellation, (b) received and distorted signal constellation, (c) equalized signal constellation, (d) equalized and quantized signal constellation

Chapter 3

Channel

This chapter introduces the ideas behind the transmitter and receiver filters for the basic digital QAM radio system, and a model of the multipath fading microwave channel. The chapter concludes with the introduction of a complex valued discrete-time baseband channel model which includes transmitter and receiver filters.

3.1 Nyquist Filters

The digital radio system consists of a transmitter and a receiver. A basic QAM radio system can be seen in Figure 2.3 [11].

The input to the transmitter is a sequence of complex valued symbols, e.g. 16-QAM symbols. The symbols are then split up into their real and imaginary (I and Q) parts before they are passed through a transmitter (pulse shaping) filter. The filter's time-continuous outputs are then added and modulated onto a carrier (up-converted from baseband to radio frequency) and transmitted over a channel. On the receiving end, the signal is down-converted back to baseband and again split up into its real and imaginary parts and passed through matched receiver filters, finally to be sampled. The idea with this radio system is that the transmitting filters in cascade with the matched receiver filters form so-called *Nyquist filters* designed for minimal inter-symbol interference. One type of transmitter filter is the square-root-raised-cosine (SQRC) filter and cascaded with another SQRC receiver filter, forms an overall raised-cosine pulse response equally distributed between transmitter and receiver. The Nyquist condition states that the minimum bandwidth of the cascaded filters is one-half of the symbol rate, e.g. if the symbol rate is 25 MHz then "brick-wall" filters of 12.5 MHz bandwidth can be used. In practical systems we don't have ideal "brick-wall" filters, so some excess bandwidth beyond this minimum requirement

is allowed.

Figure 3.1 presents an overall pulse response produced by a raised-cosine filter, this pulse satisfies the criterion for a Nyquist pulse, and if it is perfectly sampled (a timing synchronization problem), we get a discrete-time pulse response existing of only one non-zero sample and all the remaining samples are zero. This gives zero ISI during perfect channel conditons.

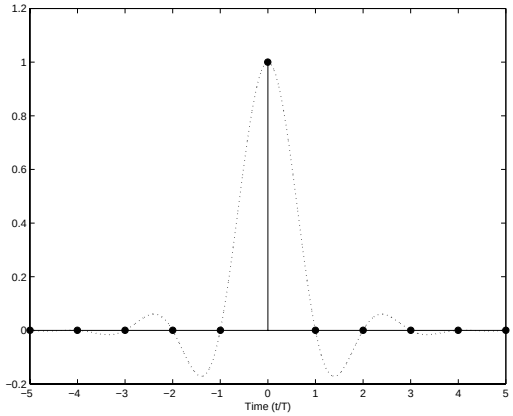

Figure 3.1: Nyquist pulse in the form of a raised-cosine pulse with 35% excess bandwidth

Now that we have looked on the the transmitter and receiver filters, it is time to have a closer look at the dispersive channel, which is the villain in this idealized radio communication scenario.

3.2 Multipath Channel

A typical scenario in a point-to-point microwave radio communication link is that the sent signal can travel multiple paths to the receiver (as was previously shown in Figure 2.4). This is due to ground reflections which arrive attenuated, phase-shifted, and delayed at the receiver, and refractions in the atmosphere due to changes in the refractive index. The atmospheric multipath propagation, also called atmospheric stratification, varies with differing humidity, temperature and pressure, so these atmospheric effects are highly dependent on the current weather conditions, and are therefore (slowly) time-varying. We can also have other reflections from obstacles for example in urban areas such as buildings, cars etc. As mentioned above, this leads to that different attenuated, phase-shifted and time-delayed version of the sent signal arriving at the receiver and causing frequency selective fading (or in time-domain, *time dispersion*), which means that the amplitude and phase characteristics of the transmission channel are frequency dependent.

In order to simulate a fading channel, we need a mathematical model of the channel. The first step is to derive a transfer function which describes the multipath fading channel.

In reality, as many as 11 rays can be present in an atmospheric multipath situation [13]. However, for practical reasons the number of rays can be reduced to three or less [13]. Mathematical models found in literature are usually based on a three-ray model or a simplification thereof. In the three-ray model we have a line-of-sight (LOS) direct ray and two reflected/refracted rays. One of these delayed rays has a delay that is sufficiently small to assume that its contribution is coherent and therefore represents flat fading rather than frequency selective fading. This three-ray model can be represented by the following impulse response [13]:

$$h(t) = a[\delta(t) - be^{j\phi}\delta(t-\tau)] \tag{3.1}$$

where a is the constant flat attenuation factor, b represents the relative amplitude of the delayed signal to the LOS one, τ is the delay of the echo, and ϕ the arbitrary phase-shift of the echo ($\delta(\cdot)$ is the well known Dirac's (generalized) delta function). These parameters should of course vary (slowly) with time, but here the time-dependence is left out since we just want to see if we can equalize a present fade at a certain time.

Let us now take the Fourier transform of $h(t)$ and study the channel's frequency response:

$$\begin{aligned} H(f) &= \mathcal{F}\{h(t)\} \\ &= a[1 - be^{-j2\pi(f-\frac{\phi}{2\pi\tau})\tau}] \\ &= a[1 - be^{-j2\pi(f-\Delta f)\tau}] \qquad (a=1) \\ &= 1 - be^{-j2\pi(f-\Delta f)\tau} \end{aligned} \tag{3.2}$$

In this equation we have introduced $\Delta f = \frac{\phi}{2\pi\tau}$, which we will refer to as the *notch displacement* from the center frequency. When varying the notch displacement, the notch will move across the channel spectrum and this is exactly what happens in practice [13]. The flat fading attenuation factor a can be set to unity because flat fading is usually compensated by an automatic gain control (AGC), and letting $a = 1$ is therefore motivated. Since we now have dropped the flat fading contribution, which was caused by one of the delayed rays, we will now refer to equation (3.2) as a two-ray model. This model has three parameters (b, Δf, and τ) and it would be convenient to reduce these to only two. One such model which assumes that the delay is constant is based on work done by William Rummler of Bell Laboratories [16]. Using measured propagation data, his model was confirmed in 1977 and it was found that for a channel with 30 MHz bandwidth, a model with 6.3 ns of delay produced accurate results for all channel conditions [13]. So,

by setting $\tau = 6.3 \times 10^{-9}$ in (3.2), we have ended up with a model, namely Rummler's model, of a frequency selective channel and this is the model that will be used in this thesis.

To find the channel zeros in the pole-zero plane, we set the Laplace transform of the impulse response to zero:

$$H(s) = \mathcal{L}\{h(t)\} = 1 - be^{j\phi}e^{-s\tau} = 0, \tag{3.3}$$

and use the complex logarithm $\log_e(\cdot)$ to solve the equation with respect to the complex s-variable:

$$\begin{aligned} \Rightarrow \quad e^{s\tau} &= be^{j\phi} \\ \Rightarrow \quad s &= \frac{1}{\tau}\log_e(be^{j\phi}), \quad b > 0 \\ &= \frac{1}{\tau}[\ln b + j(\phi + 2n\pi)], \quad n = 0, \pm 1, \pm 2, \ldots \\ &= \frac{1}{\tau}\ln b + j\frac{\phi + 2n\pi}{\tau}, \quad n = 0, \pm 1, \pm 2, \ldots \end{aligned} \tag{3.4}$$

Hence, we have an infinite amount of zeros that have an equidistant distribution with a spacing of $\frac{2\pi}{\tau}$ along a vertical axis in the complex plane. If $0 < b < 1$ this axis lies in the left half of the complex plane, if $b = 1$ the axis is equal to the imaginary axis, and if $b > 1$ it lies in the right half of the complex plane. Figure 3.2 shows the zeros in the complex plane for different b-values. The case $0 < b < 1$ is referred to as minimum phase, and $b > 1$ non-minimum phase. Minimum phase occurs when the direct LOS ray is the stronger one out of the two, and non-minimum phase when the delayed ray is the stronger one. A zero corresponds to a minima (or notch) in the magnitude frequency response and the closer the zero is to the imaginary axis, i.e. b closer to unity, the deeper the notch is. The spacing between the notches is the same as for the zeros, i.e. $\frac{2\pi}{\tau}$ rad/s $= \frac{1}{\tau}$ Hz. For the Rummler model this spacing evaluates to $\frac{1}{6.3\times 10^{-9}} \approx 160$ MHz. As in [14] we finally introduce a slightly modified channel impulse response:

$$h(t) = \begin{cases} \delta(t) - be^{j2\pi\Delta f\tau}\delta(t-\tau) & 0 < b < 1,\ \tau = 6.3 \times 10^{-9} \\ \frac{1}{b}\delta(t) - e^{j2\pi\Delta f\tau}\delta(t-\tau) & b > 1,\ \tau = 6.3 \times 10^{-9} \end{cases} \tag{3.5}$$

In where we have stated two different cases for different values of b. We have done this so we can define the notch depth in dB (the depth of the fade compared to the unfaded response) as [14]:

$$\mathrm{B_{dB}} = \begin{cases} -20\log_{10}(1-b) & 0 < b < 1 \\ -20\log_{10}(1-\frac{1}{b}) & b > 1 \end{cases} \tag{3.6}$$

Note that this modification does not effect the channel zeros of (3.4). The frequency response and notch depth for a fading channel is shown in Figure 3.3.

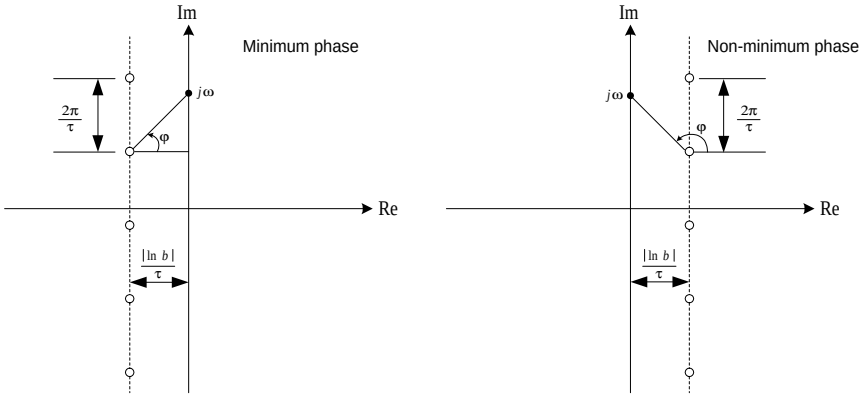

Figure 3.2: The channel zeros in the complex plane with minimum phase for $0 < b < 1$ (left) and non-minimum phase for $b > 1$ (right)

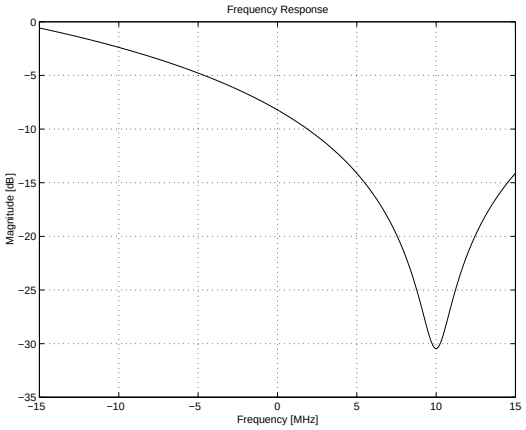

Figure 3.3: Frequency response for a minimum phase channel with notch depth $\mathrm{B_{dB}} = -20\log_{10}(1 - 0.97) = 30.5$ dB, and a notch displacement of $\Delta f = 10$ MHz

3.3 Baseband Channel Model

We want a discrete-time model of the overall channel which includes transmitter and receiver filters, and the fading channel. In order to do this, we first re-arrange the filters and channel according to Figure 3.4. In the ab-

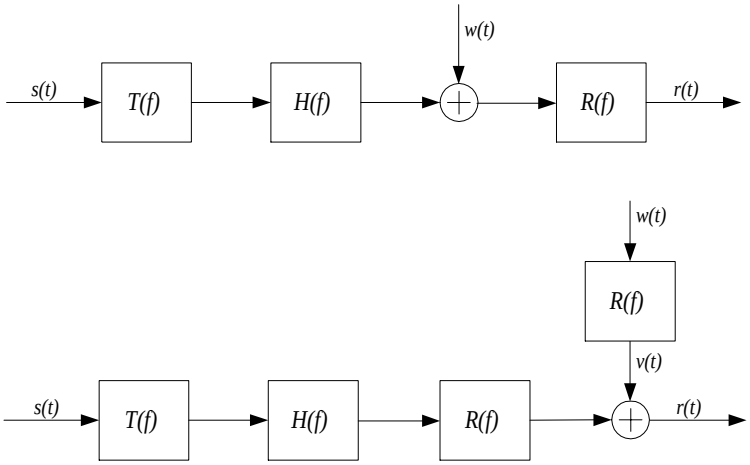

Figure 3.4: The original radio model with additive noise (top), and the equivalent scheme with re-arranged filters (bottom)

sence of noise, we now have the overall transfer function in the frequency space:

$$C(f) = T(f)H(f)R(f) = H(f)G(f) \tag{3.7}$$

where $G(f) = T(f)R(f)$ is the cascade of the SQRC filters, and therefore indicate a raised-cosine frequency response with an impulse response $g(t) = \mathcal{F}^{-1}\{G(f)\}$ in the form of a Nyquist pulse (as in Figure 3.1). Using (3.5, $0 < b < 1$) we in the time-domain get the following channel response to a Nyquist pulse:

$$\begin{aligned} c(t) &= h * g(t) = \int_{-\infty}^{\infty} h(\lambda)g(t-\lambda)\,d\lambda \\ &= \int_{-\infty}^{\infty} [\delta(\lambda) - be^{j2\pi\Delta f\tau}\delta(\lambda-\tau)]g(t-\lambda)\,d\lambda \\ &= g(t) - be^{j2\pi\Delta f\tau}g(t-\tau) \end{aligned} \tag{3.8}$$

Here $(*)$ denotes the convolution operator. Figure 3.5 shows the pulse response for a fading channel.

The baseband-equivalent received signal, $r(t)$, is a train of (by the channel) distorted pulses with the impulse response $c(t)$ delayed by integer multiples k of symbol duration T and weighted by complex-valued data symbols

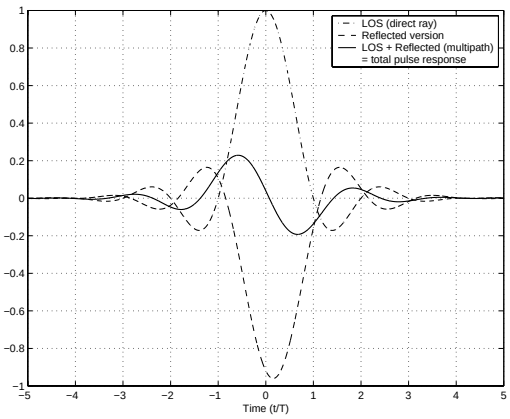

Figure 3.5: Channel pulse response for $\mathrm{B_{dB}} = 30$ dB, $\Delta f = 0$, and 35% excess bandwidth

(e.g. QAM symbols) s_k:

$$r(t) = \sum_k s_k c(t - kT) + v(t) \tag{3.9}$$

The complex source symbol sequence $\{s_k\}$ is assumed to be zero-mean, independent and identically (actually uniformly) distributed (i.i.d.), thus white uncorrelated symbols, i.e. $\mathrm{E}\{s_i s_j^\star\} = \sigma_s^2 \delta[i-j]$, where σ_s^2 is the variance of the source symbol sequence and $\delta[\cdot]$ is the Kronecker delta function. $v(t)$ is the noise process generated by filtering a complex white Gaussian and independent noise process by the receiver filter.

The corresponding discrete-time version of the received signal is obtained by sampling the received continuous-time signal at baud-rate $f_s = \frac{1}{T}$ (or, if desired, at fractions of baud-rate):

$$\begin{aligned} r_n &= r(nT) = \sum_k s_k c(nT - kT) + v(nT) \\ &= \sum_k s_k c_{n-k} + v_n, \quad n = 0, \pm 1, \ldots \end{aligned} \tag{3.10}$$

If we restrict the discrete channel response to be finite and L_c samples long and denote c_0 to be the first sample and c_{L_c-1} the last sample, then the received sequence can be the output of a baud spaced FIR filter $c(q) = c_0 + c_1 q^{-1} + \ldots + c_{L_c-1} q^{-L_c+1}$ with complex valued taps c_i, $i = 0, 1, \ldots, L_c - 1$. In other words, if we sample the time-continuous channels pulse response at symbol frequency we can simulate the frequency selective channel as a FIR filter with complex valued taps and with the source symbol sequence as input. This has now given us the desired equivalent discrete-time baseband FIR model. The FIR-model can be seen in Figure 3.6 and is described as

a tapped delay line of length L_c with taps separated by the symbol period T. The FIR channel model includes transmitter and receiver filters, and the output (received) signal has also added complex valued (and by-the-receiver filtered) white Gaussian noise with zero mean and variance σ_v^2. The FIR channel model is most likely non-minimum phase, so an equalizer model which simply inverts the channel would be unstable.

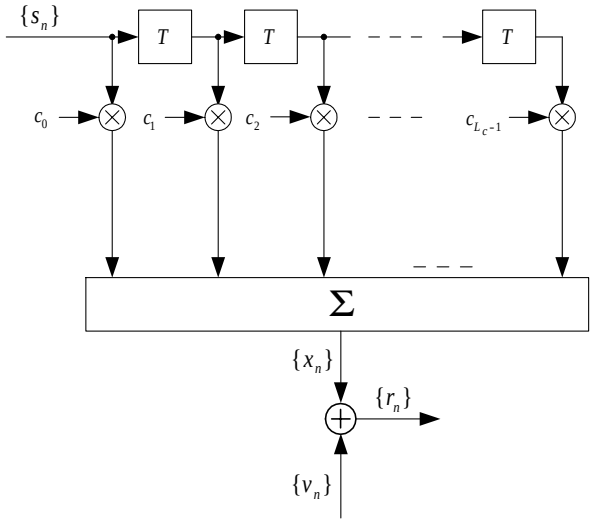

Figure 3.6: Discrete FIR channel model with baud spaced complex valued taps c_i. v_n is complex valued noise

Figure 3.7 shows the input and output constellations of the FIR channel model when the input signal is a sequence of complex symbols that are independently and equally likely drawn from a 16-QAM alphabet. As it can be seen, the impairment of the additive noise can by itself cause errors in the symbol detection (using minimum Euclidean distance). The effect of the fading channel is though so severe that the need of some sort of equalizer becomes vital since the transmitted sequence is distorted way beyond recognition and detection without any pre-equalization of the received signal is not even considered.

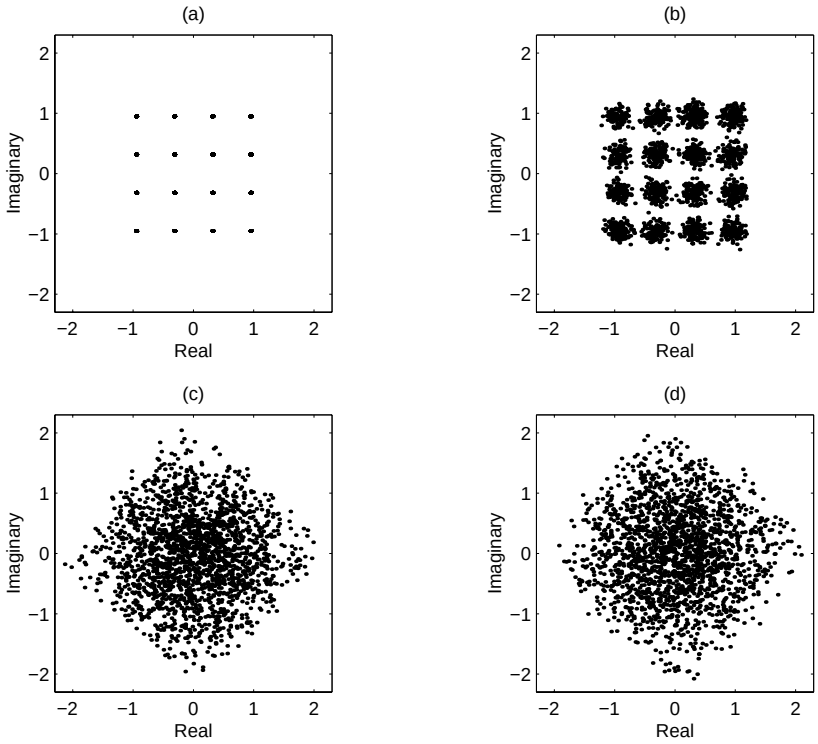

Figure 3.7: (a) Transmitted source constellation, (b) impaired by noise (no fading), (c) multipath fading present (no noise), (d) both fading and noise present

Chapter 4

Linear Equalizer

This chapter introduces the linear equalizer and various methods for updating the equalizer parameters. First we look at the structure of the complex linear equalizer and then we move on with various algorithms for equalizer updates such as; the zero-forcing and least-mean-square equalizer that imply a training sequence, the blind constant modulus algorithm and finally the blind decision-directed equalizer.

4.1 Linear Equalizer Structure

The linear equalizer (LE) is implemented as a transversal filter with complex valued taps. Figures 4.1 and 4.2 shows how a complex linear equalizer can be used for equalizing a dispersive channel with additive noise and Figure 4.3 shows a FIR equalizer with complex taps $\{f_i\}$.

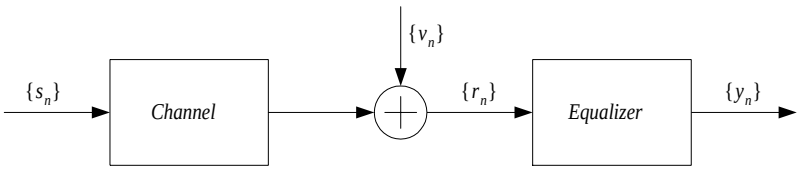

Figure 4.1: Linear equalizer

The equalized output can be expressed as[1] $y_n = c(q)f(q)s_n + f(q)v_n$, where $c(q) = c_0 + c_1q^{-1} + \ldots + c_{L_c-1}q^{-L_c+1}$, $f(q) = f_0 + f_1q^{-1} + \ldots + f_{L_f-1}q^{-L_f+1}$, $\{s_n\}$, and $\{v_n\}$ represents the channel, equalizer filter, source,

[1] q is the time delay operator such that $q^{-\ell}x[n] = x[n-\ell]$ where ℓ is an integer multiple of the period T. It is simply used, in discrete time-domain, to replace the convolution operation.

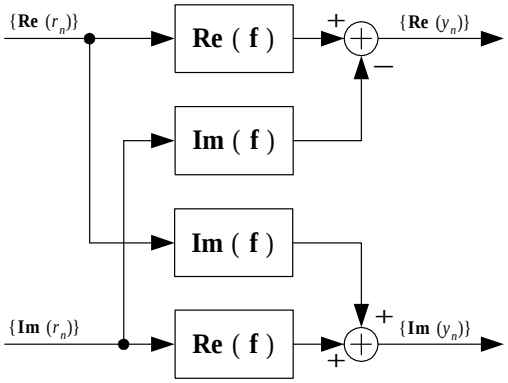

Figure 4.2: Implementation of the complex equalizer for QAM signals by dividing it into its real and imaginary parts

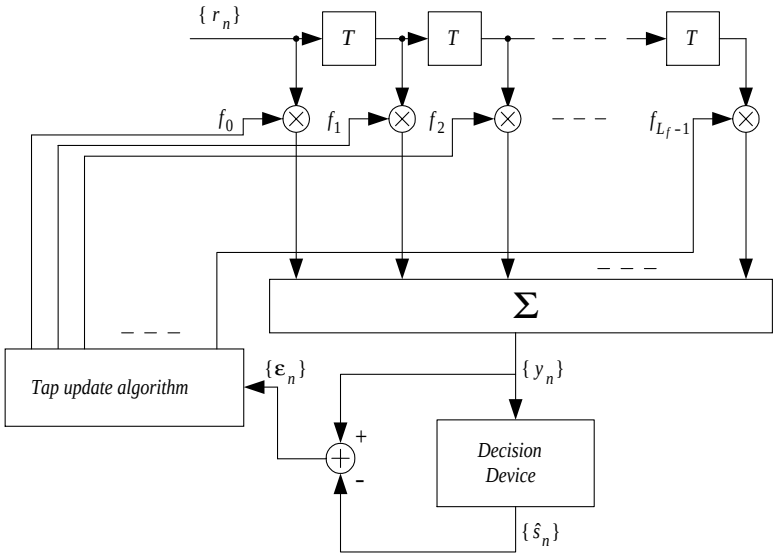

Figure 4.3: Linear equalizer in decision-directed mode implemented as a complex FIR filter of length L_f

and noise respectively. Hence, in the absence of noise, the overall channel and equalizer impulse response becomes $h^{\text{tot}}(q) = c(q)f(q)$.

The question which arises is how to go about setting the values of the equalizer taps. What we want is to reconstruct the transmitted symbols with as low probability of error as possible. There are two ways of accomplishing this: the indirect approach where the channel model is first estimated (an identification problem), and using this channel estimate, the equalizer setting is then optimized. The other (and simpler) approach is the direct one where the equalizer parameters are adjusted directly without the use of a channel estimate. It is this latter approach that is used in this thesis and to begin with we look at equalizer algorithms that use a training sequence in their initial update.

4.2 Trained Algorithms

4.2.1 Zero-forcing Equalizer

When transmitting a symbol, say s_m at time m, the best we can wish for is that the equalized output, y_n in Figure 4.1, at time $n > m$, is a time delayed version of the transmitted symbol, in other words $y_n = s_{n-\delta}$ where δ is an integer valued delay. In order to achieve this we have to have a combined channel-equalizer (or system) impulse response in the form of a delayed unit spike that appears at time δ, i.e. as shown in Figure 4.4, $h^{\text{tot}}(q) = c(q)f(q) = q^{-\delta}$. One classic and illustrative example is called the *zero-forcing equalizer* (ZFE). The ZFE technique is based on choosing the equalizer coefficients $\{f_i\}$ to "force" the combined channel and equalizer impulse response samples all to be zero, except for the sample at the center tap of the equalized response which is set to unity. In theory, by letting the number of equalizer taps increase without bound, an infinite length equalizer which, in a noise-less scenario, accomplishes zero ISI can be obtained.

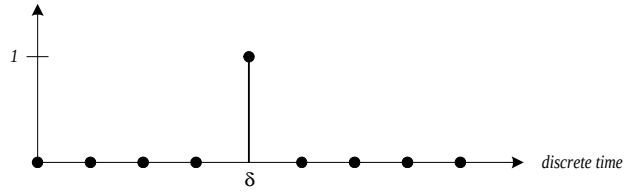

Figure 4.4: Desired combined channel-equalizer impulse response in the form of a delayed Kronecker delta function

If the input to the system is an impulse, then we want to force the equalizer output to also be an impulse, thus we require for zero ISI that the

system impulse response:

$$y_k = \begin{cases} 1 & \text{for } k = 0 \\ 0 & \text{elsewhere} \end{cases} \tag{4.1}$$

Here, $k = 0$ corresponds to the time index of the center tap in the equalized response. If we now let the equalizer have $2N + 1$ taps and modify the indexing of the equalizer taps in Figure 4.3, then can the equalizer output estimate y_k be expressed in terms of the input r_k and tap-weights f_i as follows[2]:

$$y_k = \sum_{i=-N}^{N} f_i r_{k-i} \tag{4.2}$$

Equation (4.2) yields in $2N + 1$ independent equations in terms of f_i. This limits us to $2N+1$ constraints, and therefore must equation (4.1) be modified to:

$$y_k = \begin{cases} 1 & \text{for } k = 0 \\ 0 & \text{for } -N \leq k \leq N,\ k \neq 0 \end{cases} \tag{4.3}$$

Thus the zero-forcing equalizer can force the equalizer output to be zero at limited amount of sample points on either side of the desired peak output, so samples at time $k < -N$ and $k > N$ (all depending on the length of the equalizing filter) can be non-zero. However, the zero-forcing equalizer is able to minimize the peak ISI. See Figure 4.5 and note how a 3-tap ZFE forces one sample on either side of the center tap to be zero, and how the samples beyond $k = \pm 1$ are non-zero.

Combining equations (4.2) and (4.3) gives us $2N + 1$ equations that can be solved simultaneously to acquire the equalizer tap-weights f_i. To achieve this we first define the following vectors and matrix:

$$\mathbf{f} \triangleq [f_{-N} \quad \cdots \quad f_0 \quad \cdots \quad f_N]^T \tag{4.4}$$

$$\mathbf{e}_N \triangleq [0 \quad 0 \quad \cdots \quad 1 \quad \cdots \quad 0 \quad 0]^T \tag{4.5}$$

In the $(2N + 1) \times 1$ vector $\mathbf{e}_N$ of definition (4.5), are all the elements set to zero except for the center element at index N which is set to unity.

$$\mathbf{R} \triangleq \begin{bmatrix} r_0 & r_{-1} & \cdots & r_{-N} & \cdots & r_{-2N-1} & r_{-2N} \\ r_1 & r_0 & \cdots & r_{-N+1} & \cdots & r_{-2N} & r_{-2N+1} \\ \vdots & \vdots & & & & & \\ r_N & r_{N-1} & \cdots & r_0 & \cdots & r_{-N-1} & r_{-N} \\ \vdots & \vdots & & & & & \\ r_{2N-1} & r_{2N-2} & \cdots & r_{N-1} & \cdots & r_{-2} & r_{-1} \\ r_{2N} & r_{2N-1} & \cdots & r_N & \cdots & r_{-1} & r_0 \end{bmatrix} \tag{4.6}$$

[2]The indexing of the equalizer taps is here given by equation (4.4), where f_0 is the center tap and the equalizer has N taps on either side of this center tap.

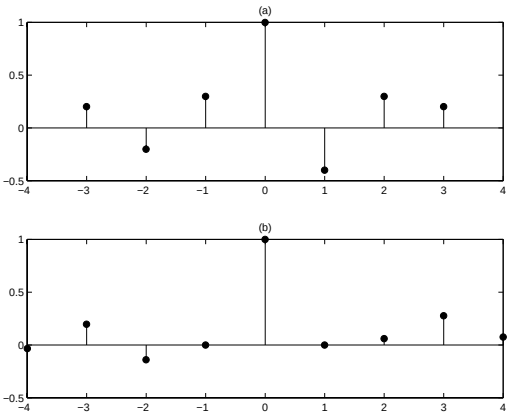

Figure 4.5: (a) Channel impulse response, (b) equalized impulse response using a 3-tap ZFE

From this we can see that if we have $2N + 1$ equalizer taps, we then need $4N + 1$ samples of the received signal to compute our ZFE. Finally, we solve the following equation system with respect to $\mathbf{f}$ to acquire the equalizer tap-weights:

$$\mathbf{Rf} = \mathbf{e}_N \tag{4.7}$$

which gives the resulting zero-forcing equalizer as:

$$\mathbf{f} = \mathbf{R}^{-1}\mathbf{e}_N \tag{4.8}$$

The ZFE has the disadvantage that the resulting equalizer may, in the presence of noise, amplify noise at frequencies where the channel spectrum has high attenuation, and for this reason it is not often used in applications involving noisy channels [1]. Although, for anyone interested, there is also iterative zero-forcing algorithms which can be initialized using a training signal (or some blind scheme) and then adapt in a *decision-directed*[3] (DD) mode. A zero-forcing algorithm is derived in [15]. But, because of the ZFE's "bad" performance for noisy channels, it will not be discussed any further in this thesis and is hereby left as an introduction to the basic idea of equalization and for the reader's education in ZFEs.

4.2.2 Least Mean Square Equalizer

While the zero-forcing equalizer minimized the ISI but did not perform well in lower signal-to-noise environments, the least mean square (LMS) equalizer minimizes the ISI without amplifying the noise and is therefore

[3] See Section 4.3.4.

more robust and gives good results also for noisy channels. There are two different modes the LMS equalizer can operate in: the training mode and the blind decision-directed mode. In the training mode, a by-the-receiver known training sequence is transmitted with certain intervals to update the equalizer, or for slowly varying point-to-point microwave channels it might only be necessary to transmit this training signal at start-up. The basic idea is to minimize the mean square error (MSE) and if we formulate the instantaneous estimation error as:

$$\varepsilon_n = y_n - s_{n-\delta} \tag{4.9}$$

where δ is some integer delay caused by the overall system, and define the mean square error (or MSE cost function) as:

$$J^{\text{mse}} \triangleq \frac{1}{2}\text{E}\{|\varepsilon_n|^2\} = \frac{1}{2}\text{E}\{|y_n - s_{n-\delta}|^2\} \tag{4.10}$$

then the resulting equalizer is the one that minimizes the MSE cost function, J^{mse}. The MSE criterion does not lead to noise enhancement since the noise is included in the criterion and a minimization of the MSE would therefore not allow noise enhancement.

If we consider the equalizer as a linear filter which using an inner-product generates the soft estimate $y_n = \tilde{\mathbf{f}}^H \mathbf{r}[n]$, where $(\cdot)^H$ is the Hermitian transpose (complex conjugate transpose), and the *tap-weight vector* $\tilde{\mathbf{f}}$ and the *regressor vector* of received symbols $\mathbf{r}[n]$ are defined as:

$$\tilde{\mathbf{f}} \triangleq [f_0^\star \quad f_1^\star \quad f_2^\star \quad \cdots \quad f_{L_f-1}^\star]^T \tag{4.11}$$

(Note how the tap-weight vector is here defined with complex conjugated tap-weights $\{f_i^\star\}$.)

$$\mathbf{r}_n = \mathbf{r}[n] \triangleq [r_n \quad r_{n-1} \quad r_{n-2} \quad \cdots \quad r_{n-L_f+1}]^T \tag{4.12}$$

Then the MSE cost function becomes:

$$\begin{aligned} J^{\text{mse}} &\triangleq \frac{1}{2}\text{E}\{|\varepsilon_n|^2\} \\ &= \frac{1}{2}\text{E}\{|y_n - s_{n-\delta}|^2\} \\ &= \frac{1}{2}\text{E}\{|\tilde{\mathbf{f}}^H \mathbf{r}[n] - s_{n-\delta}|^2\} \end{aligned} \tag{4.13}$$

In this equation $s_{n-\delta}$ is the transmitted symbol at time $n-\delta$ and this formulation of the MSE cost requires that $s_{n-\delta}$ is known at the receiver, and it is here the training sequence is required (if the transmitted symbols would be known by the receiver without the use of a training sequence, then the need of a communication system would be defeated since the receiver

of the information is obviously both psychic and clairvoyant). The training sequence is on forehand stored in the receiver and Figure 4.6 shows how a training sequence of length L can be organized in a transmission sequence. The length of the training sequence is also a design parameter, the longer training the better error-rate performance is obtained. Now that we have

Figure 4.6: Organization of the training sequence in a transmission sequence

introduced the training sequence, we can continue with how to choose an optimal equalizer setting which minimizes the MSE cost function.

If we rewrite and multiply out the MSE cost function we get:

$$\begin{aligned}
J^{\text{mse}} &= \frac{1}{2}\mathrm{E}\{|\varepsilon_n|^2\} = \frac{1}{2}\mathrm{E}\{\varepsilon_n\varepsilon_n^\star\} \\
&= \frac{1}{2}\mathrm{E}\{(\tilde{\mathbf{f}}^H\mathbf{r}[n] - s_{n-\delta})(\tilde{\mathbf{f}}^H\mathbf{r}[n] - s_{n-\delta})^\star\} \\
&= \frac{1}{2}\mathrm{E}\{(\tilde{\mathbf{f}}^H\mathbf{r}[n] - s_{n-\delta})(\mathbf{r}^H[n]\tilde{\mathbf{f}} - s_{n-\delta}^\star)\} \\
&= \frac{1}{2}\mathrm{E}\{\tilde{\mathbf{f}}^H\mathbf{r}[n]\mathbf{r}^H[n]\tilde{\mathbf{f}} - \tilde{\mathbf{f}}^H\mathbf{r}[n]s_{n-\delta}^\star \\
&\quad -s_{n-\delta}\mathbf{r}^H[n]\tilde{\mathbf{f}} + s_{n-\delta}s_{n-\delta}^\star\} \\
&= \frac{1}{2}\tilde{\mathbf{f}}^H\overbrace{\mathrm{E}\{\mathbf{r}[n]\mathbf{r}^H[n]\}}^{\mathbf{R}_n}\tilde{\mathbf{f}} - \frac{1}{2}\tilde{\mathbf{f}}^H\overbrace{\mathrm{E}\{\mathbf{r}[n]s_{n-\delta}^\star\}}^{\mathbf{h}_n} \\
&\quad -\frac{1}{2}\underbrace{\mathrm{E}\{s_{n-\delta}\mathbf{r}^H[n]\}}_{\mathbf{h}_n^H}\tilde{\mathbf{f}} + \frac{1}{2}\underbrace{\mathrm{E}\{s_{n-\delta}s_{n-\delta}^\star\}}_{\sigma_s^2} \\
&= \frac{1}{2}\tilde{\mathbf{f}}^H\mathbf{R}_n\tilde{\mathbf{f}} - \frac{1}{2}\tilde{\mathbf{f}}^H\mathbf{h}_n - \frac{1}{2}\mathbf{h}_n^H\tilde{\mathbf{f}} + \frac{1}{2}\sigma_s^2
\end{aligned} \tag{4.14}$$

Note that $\mathbf{r}[n]$ and $s_{n-\delta}$ are stochastic quantities, but $\tilde{\mathbf{f}}$ is not. $\mathbf{R_n}$ is the received signals covariance matrix and σ_s^2 is the (zero-mean) source variance. Since J^{mse} is a positive quadratic function of $\tilde{\mathbf{f}}$, there is only one minimum which therefore is global. It can be shown that the gradient with respect to $\tilde{\mathbf{f}}$ of the MSE cost evaluates to:

$$\nabla_{\tilde{\mathbf{f}}}J^{\text{mse}} = \mathbf{R}_n\tilde{\mathbf{f}} - \mathbf{h}_n \tag{4.15}$$

and if $\nabla_{\tilde{\mathbf{f}}}J^{\text{mse}}$ is set equal to the null-vector, the (in MSE sense) optimal solution becomes:

$$\tilde{\mathbf{f}}_{\text{opt}} = \mathbf{R}_n^{-1}\mathbf{h}_n \tag{4.16}$$

This is the solution to the normal equations and is solvable if the matrix $\mathbf{R}_n$ is non-singular, i.e. $\det[\mathbf{R}_n] \neq 0$. The normal equations can be solved directly using matrix inversion or alternatively an iterative gradient method, e.g. *stochastic gradient descent* (SGD) method. The cost function is quadratic and has a single minimum located where its derivative with respect to the equalizer parameters is equal to zero. To reach this point, we have to iteratively move using small steps in the negative gradient direction. So, an equalizer update algorithm can be the one that minimizes the MSE cost function using the following SGD method:

$$\tilde{\mathbf{f}}[n+1] = \tilde{\mathbf{f}}[n] - \mu \nabla_{\tilde{\mathbf{f}}} J^{\text{mse}} \tag{4.17}$$

where μ is a (small) step-size. We can now use the gradient we computed in (4.15) to get:

$$\tilde{\mathbf{f}}[n+1] = \tilde{\mathbf{f}}[n] - \mu(\mathbf{R}_n \tilde{\mathbf{f}}[n] - \mathbf{h}_n) \tag{4.18}$$

The covariance matrix $\mathbf{R}_n$ and the vector $\mathbf{h}_n$ are expectation values and these can be approximated by their instantaneous estimates:

$$\hat{\mathbf{R}}_n = \mathbf{r}[n]\mathbf{r}^H[n] \quad \text{and} \quad \hat{\mathbf{h}}_n = \mathbf{r}[n]s^\star_{n-\delta} \tag{4.19}$$

Using the above estimates together with equation (4.18), we get:

$$\begin{aligned} \tilde{\mathbf{f}}[n+1] &= \tilde{\mathbf{f}}[n] - \mu \mathbf{r}[n] \underbrace{(\mathbf{r}^H[n]\tilde{\mathbf{f}}[n] - s^\star_{n-\delta})}_{\varepsilon^\star_n} \\ &= \tilde{\mathbf{f}}[n] - \mu \mathbf{r}[n]\varepsilon^\star_n \end{aligned} \tag{4.20}$$

If we now take the complex conjugate of both sides and re-define the equalizer tap-weight vector with its conjugate:

$$\mathbf{f} \triangleq \tilde{\mathbf{f}}^\star = [f_0 \quad f_1 \quad f_2 \quad \cdots \quad f_{L_f-1}]^T \tag{4.21}$$

then we can re-write the iterative equalizer update algorithm as:

$$\begin{aligned} \mathbf{f}[n+1] &= \mathbf{f}[n] - \mu \varepsilon_n \mathbf{r}^\star[n] \\ &= \mathbf{f}[n] - \mu(\mathbf{f}[n]^T \mathbf{r}[n] - s_{n-\delta})\mathbf{r}^\star[n] \\ &= \mathbf{f}[n] - \mu(y_n - s_{n-\delta})\mathbf{r}^\star[n] \end{aligned} \tag{4.22}$$

The above algorithm is called the *least mean square* (LMS) algorithm. LMS is, due to its simplicity, very popular and one of the most studied and widely used algorithms in trained equalizer applications.

The time delay δ in the LMS algorithm is a design parameter and it is important that it is properly pre-selected. A solution that has worked well in simulations is that the total time delay is set to half the equalizer length, i.e. $\delta = \lfloor \frac{L_f}{2} \rfloor$.

A necessary, but not sufficient, step-size condition to assure stability of the LMS-algorithm is [19]:

$$0 < \mu < \frac{1}{\lambda_{\max}} \tag{4.23}$$

where $\lambda_{\max}$ is the maximum eigenvalue of the covariance matrix $\mathbf{R}_n$. A large step-size leads to faster convergence but the final setting will have a somewhat large excess error (excess MSE and/or ISI [4]). On the other hand, a small step-size will lead to a better setting and low excess error but have slow convergence. Therefore is the choice of step-size a trade-off between convergence speed and fine equalizer settings. An adaptive step-size $\mu = \mu[n]$ that has large values in the beginning and tend to zero when the equalizer adaptation is in the close neighborhood of the "optimal" solution, would give satisfying convergence speed and low excess error levels.

4.3 Blind Algorithms

The use of a training sequence is sometimes not motivated because it consumes bandwidth. The reason being is that instead for transmitting *unknown* source data, *known* data is transmitted in training mode and therefore training consumes bandwidth efficiency. There are equalization schemes that do not require training sequences, so called *blind* equalization algorithms.

One of the most common blind algorithms that is widely used in practical implementations is the *constant modulus algorithm* (CMA) [8].

4.3.1 Constant Modulus Algorithm

Blind equalization is a way of estimating "unknown"[5] source symbols $\{s_n\}$, without use of any known training symbols, from the received sequence $\{\mathbf{r}_n\}$, also called *regressor* sequence.

$$\mathbf{r}_n = \mathbf{r}[n] \triangleq [r_n \quad r_{n-1} \quad r_{n-2} \quad \cdots \quad r_{n-L_f+1}]^T \tag{4.24}$$

To motivate the introduction of the constant modulus algorithm, we again consider the equalization problem as a linear filter $\mathbf{f}$ of length L_f:

$$\mathbf{f} \triangleq [f_0 \quad f_1 \quad f_2 \quad \cdots \quad f_{L_f-1}]^T \tag{4.25}$$

which, using an inner-product, generates a symbol estimate $y_n = \mathbf{f}^T\mathbf{r}[n]$ and we desire that $y_n \approx s_{n-\delta}$ for some fixed (integer) system delay δ.

[4] These quantities are defined in Section 6.1.

[5] We do have knowledge of the source alphabet constellation which is a key to the solution of the blind equalizer problem.

Once again consider a typical equalizer update algorithm using a stochastic gradient descent (SGD) method:

$$\mathbf{f}[n+1] = \mathbf{f}[n] - \mu \nabla_{\mathbf{f}} J \tag{4.26}$$

The SGD algorithm uses a steepest descent method (i.e. minimizing by moving in the $-\nabla_{\mathbf{f}} J$ direction). Typically an instantaneous estimate of the gradient is used to form a more practical descent algorithm for updating the equalizer parameters. Therefore, the following LMS-like algorithm is introduced:

$$\mathbf{f}[n+1] = \mathbf{f}[n] - \mu \varepsilon_n \mathbf{r}^{\star}[n] \tag{4.27}$$

In this equation, ε_n is the so called *error function*. In a situation with a training update, $\varepsilon_n = y_n - s_{n-\delta}$, and in a situation of a *decision-directed*[6] (DD) update, $\varepsilon_n = y_n - \hat{s}_n$, where $\hat{s}_n$ is the symbol estimate generated by the decision device. In some situations the estimates $\{\hat{s}_n\}$ are not very reliable, e.g. during cold start-up, and therefore is a more reliable error signal ε_n needed. Therefore consider applying a memoryless non-linearity $g(\cdot)$, as in Figure 4.7, to the equalizer output to form the error [7]:

$$\varepsilon_n = y_n - g(y_n) \tag{4.28}$$

This error signal satisfies the criteria for *blind* equalization since ε_n is constructed solely from the received signal and does therefore not require any training symbols.

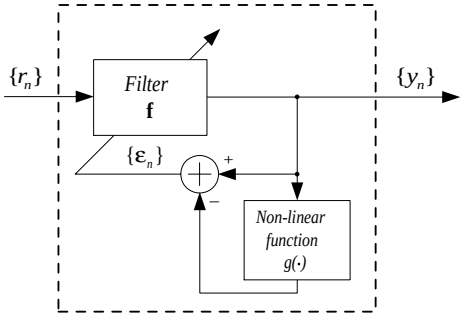

Figure 4.7: Adaptive blind equalization using a stochastic gradient descent algorithm

As mentioned in Section 2.4, the receiver must not only remove ISI but also compensate for a phase offset. According to [18], practical considerations motivate the decoupling of blind equalization from carrier recovery. When the source is differentially encoded, knowledge of the absolute phase

[6] See Section 4.3.4.

is not required for symbol detection since in differentially encoded signals the information is carried in the phase transitions. This lead Godard [6] to study carrier-phase independent algorithms based only on the signal modulus $|y_n|$ and he introduced the following cost function that (when minimized) penalizes the dispersion:

$$\begin{aligned} J_p^{\text{Godard}} &\triangleq \frac{1}{2p}\mathrm{E}\{(|y_n|^p-\gamma)^2\} \\ &= \frac{1}{2p}\mathrm{E}\{(|\mathbf{f}^T\mathbf{r}[n]|^p-\gamma)^2\} \end{aligned} \tag{4.29}$$

where γ (also known as the *dispersion constant* or *Godard radius*[7]) is a real constant and its choice is based on the source alphabet and the integer p. The desired phase-independence is acquired since:

$$|y| = |e^{j\phi}y| \;\Rightarrow\; J_p^{\text{Godard}}(y) = J_p^{\text{Godard}}(e^{j\phi}y), \quad \text{for any phase-shift } \phi. \tag{4.30}$$

By setting $\gamma = \mathrm{E}\{|s_n|^{2p}\}/\mathrm{E}\{|s_n|^p\}$, it was shown that the existence of local minima of J_p^{Godard} that perfectly equalizes the channel was ensured [6]. When $p=2$, (4.29) is referred to as the CM *cost* or the CM *criterion*, and is denoted J^{cm} [7]:

$$\begin{aligned} J^{\text{cm}} &\triangleq \frac{1}{4}\mathrm{E}\{(|y_n|^2-\gamma)^2\} \\ &= \frac{1}{4}\mathrm{E}\{(|\mathbf{f}^T\mathbf{r}[n]|^2-\frac{\mathrm{E}\{|s_n|^4\}}{\sigma_s^2})^2\} \end{aligned} \tag{4.31}$$

The CM cost penalizes the dispersion of the squared output modulus $|y_n|^2$ away from the constant $\mathrm{E}\{|s_n|^4\}/\sigma_s^2$. Since CM is a phase-independent criterion for blind equalization which minimizes dispersion, intuitively it would only work well with *CM sources*, i.e. alphabets of equal symbol magnitude (e.g. M-PSK). Remarkably, the CM criterion works almost as well for *non*-CM sources (e.g. M-QAM, where $M>4$) [7].

To acquire the equalizer update algorithm (4.27), leading to a LMS-like descent of the J^{cm} function (4.31), we first need to compute the gradient of J^{cm} with respect to $\mathbf{f}$. It is shown in [17] that the gradient evaluates to:

$$\nabla_{\mathbf{f}} J^{\text{cm}} = \mathrm{E}\{y_n(|y_n|^2-\gamma)\mathbf{r}^\star[n]\} \tag{4.32}$$

As for the LMS case, we can use an instantaneous estimate of the above gradient, i.e. drop the expectation operator. The equalizer update algorithm, namely CMA, will then follow as:

$$\begin{aligned} \mathbf{f}[n+1] &= \mathbf{f}[n] - \mu\nabla_{\mathbf{f}} J^{\text{cm}} \\ &= \mathbf{f}[n] - \mu \underbrace{y_n(|y_n|^2-\gamma)}_{\varepsilon_n^{\text{cma}}}\mathbf{r}^\star[n] \end{aligned} \tag{4.33}$$

[7] See Section A.1 for values of γ in M-QAM constellations.

where $\varepsilon_n^{\text{cma}}$ is the instantaneous CM error function in the CM algorithm. The baud spaced CMA tap-update in (4.33) is usually initialized with a normalized single-spike somewhere in the center of equalizer tap vector, e.g. $f_{\lfloor L_f/2 \rfloor} = 1$. This initial setting is nothing else but an all-pass filter and corresponds somewhat to the choice of delay for the LMS equalizer. Figure 4.8 shows an example plot of the CM cost function and the instantaneous error in CMA for 4-PAM[8] signaling.

An useful guideline for the step-size in CMA (for $\sigma_s = 1$) is [7]:

$$0 < \mu \leq \frac{2}{(\kappa_g - \kappa_s)\lambda_{\max}} \tag{4.34}$$

where κ_g and κ_s are the *normalized source kurtosises*[9] for a Gaussian source and the actual source respectively. Here is $\lambda_{\max}$ the maximum eigenvalue of the Hermitian matrix $\mathbf{C}^H\mathbf{C}$ where $\mathbf{C}$ is the BS channel convolution matrix formed by the channel impulse response samples [7]:

$$\mathbf{C} \triangleq \begin{bmatrix} c_0 & \cdots & c_{L_c-1} & & \\ & \ddots & & \ddots & \\ & & c_0 & \cdots & c_{L_c-1} \end{bmatrix} \tag{4.35}$$

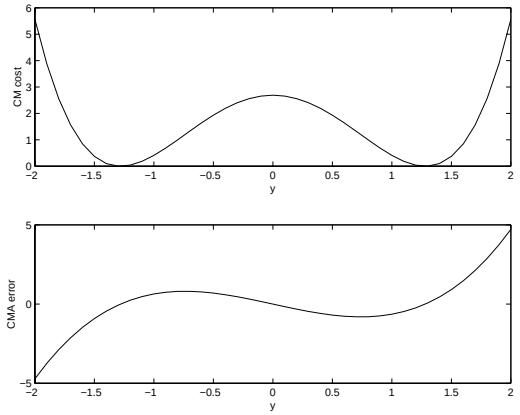

Figure 4.8: Example of the cost function $J^{\text{cm}}(y)$ (top), and the error function $\varepsilon^{\text{cma}}(y)$ (bottom) for 4-PAM signaling

The conditions assuring global convergence to a perfect blind setting of a T-spaced equalizer updated using CMA are [8]:

(i) No additive channel noise

[8] In pulse amplitude modulation (PAM) signaling the symbols are drawn from the alphabet $\{\pm\Delta, \pm 3\Delta, \pm 5\Delta, \dots\}$, i.e. equidistant real-valued symbols.

[9] See Section A.2.

(ii) Infinite impulse response (IIR) equalizer

(iii) No nulls in the channel frequency response, i.e. no FIR channel zeros on the unit circle

(iv) Zero-mean, independent (and circularly symmetric if complex valued) finite alphabet source with *sub-Gaussian kurtosis*.

Note that the conditions do not include a restriction to a CM-source. But the drawback of the baud-spaced equalizer is that, to be able to guarantee global convergence when using CMA, we need an infinite long equalizer, i.e. IIR equalizer. Some of the conditions will of course be violated, e.g. noise and IIR equalizer, but the equalizer will still be quite robust and perform quite well in a sub-optimal setting [7]. Figure 4.9 shows the importance of a FIR equalizer having an appropriate length, and it is evident that the 15-tap and 25-tap CMA equalizers here perform much better than the shorter ones. Figure 4.10 shows the equalized constellations (after 500,000 iterations) for these two equalizers.

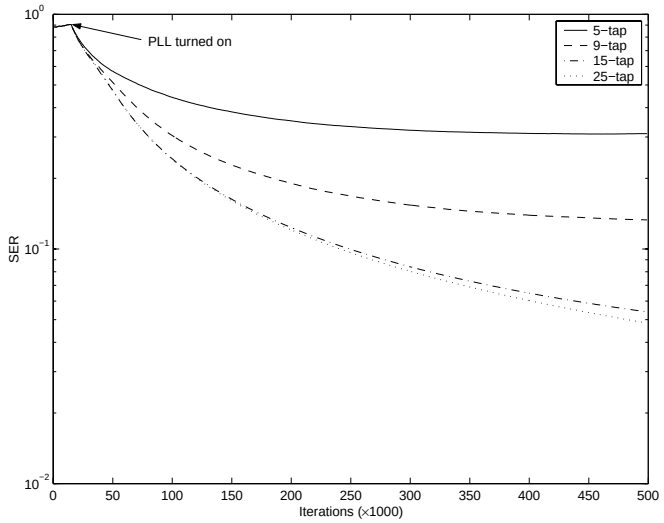

Figure 4.9: Example of the SER history for various-length CMA equalizers. The channel is strongly fading with additive noise

4.3.2 Fractionally Spaced CMA

Later discoveries have shown that a *fractionally-spaced equalizer* (FSE) has properties such as: less sensitivity to clock synchronization, matched filter abilities, reduced noise enhancement, etc [5]. Another feature that has recently been discovered is that a finite length FSE (FIR FSE), under ideal conditions, can perfectly equalize a FIR channel [2].

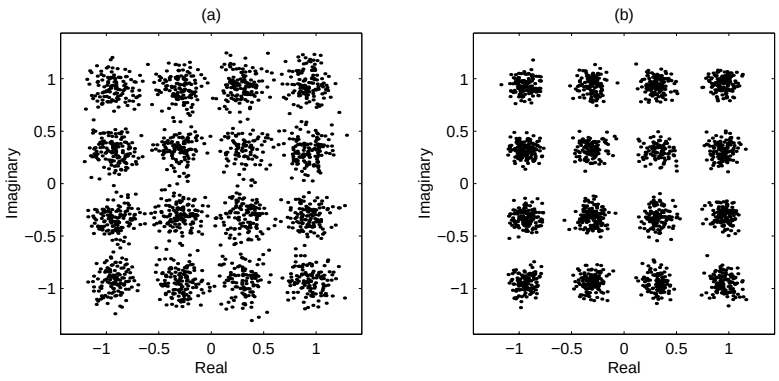

Figure 4.10: Last 2,000 symbols of the equalized constellations for (a) the 15-tap, and (b) 25-tap CMA equalizers. The channel is that of Figure 4.9

In comparison to the previously discussed baud-spaced equalizers (BSE), a T/L-spaced equalizer over-samples the received signal L times and the taps of the equalizer FIR filter are T/L-spaced. We will focus here on $T/2$-spaced equalizers, but the results are extendible to the more general T/L-spaced case.

Figure 4.11 shows a block diagram of the time-continuous $T/2$-spaced baseband model and Figure 4.12 shows the equivalent discrete multirate model. The multirate model is derived in [8] using $T/2$-spaced samples of $c(t)$ and $v(t)$. The multirate model in Figure 4.12 uses the discrete-

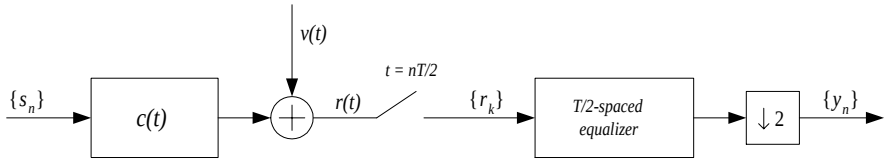

Figure 4.11: Continuous-time $T/2$-spaced baseband model

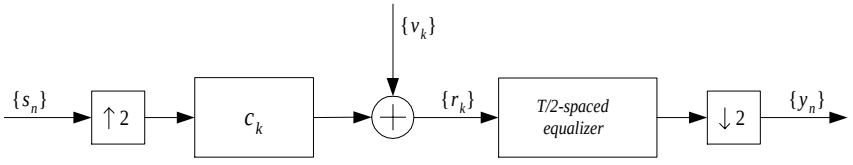

Figure 4.12: Equivalent multirate model

time fractionally-spaced (FS) channel coefficients $\{c_k\} = \{c(k\frac{T}{2})\}$ and the

discrete-time stochastic noise process $\{v_k\} = \{v(k\frac{T}{2})\}$. It can be interesting to study the BS overall channel-equalizer impulse response of the FS system. Thus, to acquire the BS response, convolve the FS channel impulse response with the FS equalizer impulse response, then the BS overall channel-equalizer impulse response (in the absence of noise) is obtained by decimating the result of the FS convolution 2-times, i.e. $h^{\text{tot}}(q) = [c(\tilde{q})f(\tilde{q})]_{\downarrow 2}$, where q^{-1} and $\tilde{q}^{-1}$, are the BS unit delay and the two-times FS unit delay operators, respectively.

In the baud-spaced case it is important to sample at the optimal time-instants, otherwise it will lead to ISI. Now we sample the channel response at twice the baud-rate, as shown in Figure 4.13. Fractional sampling is usually implemented to ease the baud-spaced clock synchronization problem. The

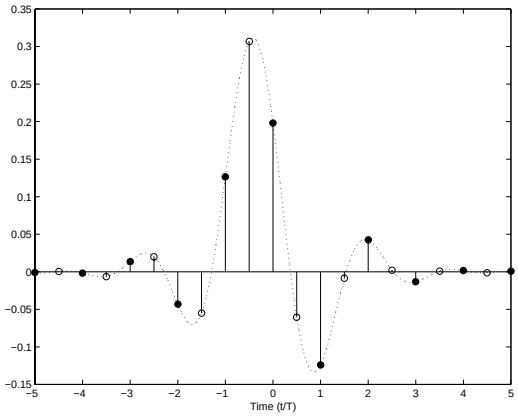

Figure 4.13: Two-times fractional sampling of a fading channel pulse response

gradient descent algorithm for FS-CMA becomes the same as for the baud-spaced equalizer:

$$\begin{aligned} \mathbf{f}[n+1] &= \mathbf{f}[n] - \mu\varepsilon_n^{\text{cma}}\mathbf{r}^\star[n] \\ &= \mathbf{f}[n] - \mu y_n(|y_n|^2 - \gamma)\mathbf{r}^\star[n] \end{aligned} \tag{4.36}$$

and is updated on each baud interval n. But with the difference being the equalizer tap-elements in the vector $\mathbf{f}$ are FS and the two-times FS regressor vector $\mathbf{r}[n]$ at baud-instant n is:

$$\mathbf{r}[n] \triangleq [r_{2n} \quad r_{2n-1} \quad r_{2n-2} \quad \cdots \quad r_{2n-L_f+1}]^T \tag{4.37}$$

where L_f is the number of FS equalizer taps. Thus, the baud-spaced symbol estimates are generated as $y_n = \mathbf{f}^T\mathbf{r}[n]$. The update is usually initialized with a normalized single- or double-spike (two adjacent taps set to $\frac{1}{\sqrt{2}}$)

somewhere in the center of the equalizer tap vector. The double-spike initialization has a low-pass characteristic which is shared by the transmitter's pulse-shaping filter.

The step-size requirement in the two-times FS-CMA (for $\sigma_s = 1$) is [7]:

$$0 < \mu \leq \frac{2}{(\kappa_g - \kappa_s)\lambda_{\max}} \tag{4.38}$$

which is the same as for the BS-CMA of (4.34), but with the difference being that $\mathbf{C}$ in the Hermitian matrix $\mathbf{C}^H\mathbf{C}$ is the FS channel convolution matrix formed by the even-length and the two-times FS channel impulse response:

$$\mathbf{C} \triangleq \begin{bmatrix} c_1 & \cdots & c_{L_c-1} & & \\ c_0 & \cdots & c_{L_c-2} & & \\ & c_1 & \cdots & c_{L_c-1} & \\ & c_0 & \cdots & c_{L_c-2} & \\ & & \ddots & \ddots & \\ & & c_1 & \cdots & c_{L_c-1} \\ & & c_0 & \cdots & c_{L_c-2} \end{bmatrix} \tag{4.39}$$

where the odd rows $\{1, 3, \dots\}$ contain the odd FS samples $\{c_1, c_3, \dots, c_{L_c-1}\}$ and the even rows, the even FS samples $\{c_0, c_2, \dots, c_{L_c-2}\}$.

The conditions assuring global convergence to a perfect blind setting of a $T/2$-spaced FSE updated using CMA are [8]:

(i) No channel noise

(ii) Equalizer time span matching or exceeding that of the FIR channel

(iii) No common roots in the polynomials that are formed by the even and odd samples of an even-length $T/2$-spaced sampled FIR channel impulse response $\mathbf{c} = [c_0 \quad c_1 \quad \cdots \quad c_{L_c-1}]$, i.e. $c_{\text{even}}(z^{-1}) = c_0 + c_2 z^{-1} + \dots$ and $c_{\text{odd}}(z^{-1}) = c_1 + c_3 z^{-1} + \dots$ can not share any common roots.

(iv) Zero-mean, independent (and circularly symmetric if complex valued) finite alphabet source with *sub-Gaussian kurtosis*[10]

Note that the FSE-CMA conditions as for the BSE-CMA conditions do not include a restriction to a CM-source, and even if some of the conditions are violated, the equalizer will perform well and be quite robust [7].

[10] See Section A.2.

4.3.3 Adaptive Complex Gain Control

When using phase-independent algorithms such as CMA, we also need to implement some adaptive phase-tracking algorithms. The channel model which was previously introduced generates impulse responses with imaginary parts of the same size as its real parts, i.e. it will introduce a phase-offset.

Figure 4.14 shows how an *adaptive complex gain control* (ACGC) can be implemented. The equalizer output is multiplied by the complex valued

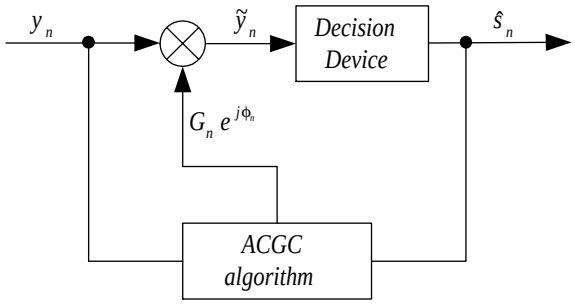

Figure 4.14: Adaptive complex gain control

output from the ACGC:

$$\tilde{y}_n = G_n \exp(j\phi_n) y_n = G_n \exp(j\phi_n) \mathbf{f}^T \mathbf{r}[n] \tag{4.40}$$

The ACGC compensates for lost gain G_n and phase ϕ_n on every iteration n. The CM criterion has a "built-in" power recovery, therefore the power recovery factor G_n is not essential. But if it is included, we gain a faster convergence rate when using CMA. However, the carrier-phase tracking is most essential otherwise will the equalized signal constellation be rotated from its original position.

The power adjustment factor in equation (4.40) can be adjusted by the simple adaptive gain control algorithm introduced in [4]:

$$G[n+1] = G[n] + \lambda(\sigma_s^2 - |\tilde{y}_n|^2) \tag{4.41}$$

where λ is the (small) adaptation gain factor, and σ_s the variance of the zero-mean, independent source symbol sequence.

The adaptation of the carrier-phase, which actually is a phase-locked-loop (PLL), requires some phase-error signal. One simple, and easily implementable using discrete logics, phase-error signal that gives good performance for QAM, is the one suggested in [10]:

$$\varepsilon_n^{\phi} = \mathrm{sgn}(\Re\{\varepsilon_n^{\mathrm{dd}}\})\mathrm{sgn}(\Im\{\tilde{y}_n\}) - \mathrm{sgn}(\Im\{\varepsilon_n^{\mathrm{dd}}\})\mathrm{sgn}(\Re\{\tilde{y}_n\}) \tag{4.42}$$

where $\varepsilon_n^{\mathrm{dd}}$ is the decision error $\varepsilon_n^{\mathrm{dd}} = \tilde{y}_n - \hat{s}_n$, $\mathrm{sgn}(\cdot)$ the sign-operator, $\Re\{\cdot\}$ and $\Im\{\cdot\}$ are the real- and imaginary-part operators, respectively.

We can use a simple first-order phase-tracking loop for adapting the phase:

$$\phi[n+1] = \phi[n] + \alpha\varepsilon_n^{\phi} \tag{4.43}$$

where α is the (small) adaptive gain factor.

There are some problems related to phase-tracking some of which are described in [4]. Typical problems are: PLL spinning, large constellation rotation and high sensitivity to phase jitter. These problems occur since the algorithm depends on correct past decisions. [4] suggest that PLL spinning may be solved by reducing the large adaptation noise, i.e. letting the equalizer algorithm (CMA) converge as fast as possible while the PLL is turned off. At convergence of CMA the PLL is turned on to rotate the constellation into place, and at this point the PLL should not spin anymore. However, if the constellation has to be rotated too far when the PLL first is turned on, spinning may still occur. The suggested solution in [4] is to make a rough estimate of the initial value $\phi[0]$ by counting the number of points that lie inside certain "templates" of the equalized constellation. Refer to [4] for further discussion regarding this solution.

Some changes are also needed to be done to the original equalizer tap-update algorithms now when the ACGC algorithms are to be implemented. The modified tap-update algorithm becomes [4]:

$$\mathbf{f}[n+1] = \mathbf{f}[n] - \mu\varepsilon_n\mathbf{r}^{\star}[n]G_n\exp(-j\phi_n) \tag{4.44}$$

The above algorithm applies to CMA and to the *decision-directed* update (which is discussed in the following section) if the initial convergence was accomplished by using CMA.

4.3.4 Decision-Directed Blind Adaptive Equalization

After successful convergence of a blind equalization algorithm such as CMA or after that training using the LMS algorithm has completed and we have a good enough equalizer setting, i.e. the symbol estimates are with high probability equal to the true symbols, then we can transfer the equalizer parameter update to a *decision-directed* (DD) mode. In this mode the *hard*[11] symbol estimates and hopefully correct ones are used in the parameter update equations instead for the training symbols, i.e. the true symbols $\{s_n\}$ are replaced by their estimates $\{\hat{s}_n\}$. This way we can continue to adapt the equalizer

[11] The equalizer filter outputs (or *soft* estimates) are passed onto a *decision device* (or *quantizer*) that computes a *hard* estimate (or *decision*) using the minimum Euclidean distance (or nearest neighbor).

in the communication (or unknown symbol transmission) mode, and even though training has ended the equalizer can continue to minimize the MSE, suppress ISI and/or "track" small channel variations. By transferring to DD mode after initial convergence accomplished by CMA, it will further minimize the excess MSE and ISI, and by there lead to a better equalizer setting. The DD equalizer is attractive because of its similarities to LMS and its qualities; convergence, simplicity, tracking ability, etc.

In DD mode is the error function generated as:

$$\varepsilon_n^{\mathrm{dd}} = y_n - \hat{s}_n \tag{4.45}$$

Notice that if $\hat{s}_n = s_{n-\delta}$, then this is identical to the LMS case. So by using this scheme, we can adapt the equalizer without the presence of a previously known training signal. According to [12] the DD equalizer converges to the optimal setting when initialized at an *open-eye*[12] setting. It is enough for the equalizer to be set to a marginally open-eye setting for a successful switch to DD mode. As long as the requirement $\hat{s}_n = s_{n-\delta}$ occurs often enough, we converge to a optimal equalizer setting. So in DD mode, perfect estimation of the source symbols is not needed all the time for the equalizer to converge to an optimal setting.

The cost function (mean squared error) to be minimized in DD mode is:

$$\begin{aligned} J^{\mathrm{dd}} &\triangleq \frac{1}{2}\mathrm{E}\{|\varepsilon_n^{\mathrm{dd}}|^2\} \\ &= \frac{1}{2}\mathrm{E}\{|y_n - \hat{s}_n|^2\} \end{aligned} \tag{4.46}$$

and after minimizing with respect to $\mathbf{f}$, and dropping the expectation operator, the equalizer update algorithm becomes:

$$\begin{aligned} \mathbf{f}[n+1] &= \mathbf{f}[n] - \mu\varepsilon_n^{\mathrm{dd}}\mathbf{r}^\star[n] \\ &= \mathbf{f}[n] - \mu(y_n - \hat{s}_n)\mathbf{r}^\star[n] \end{aligned} \tag{4.47}$$

Figure 4.15 displays a linear equalizer operating in DD mode after initial convergence using a training sequence.

[12] Open-eye is a state in which the equalizer is set to give appropriately low SER.

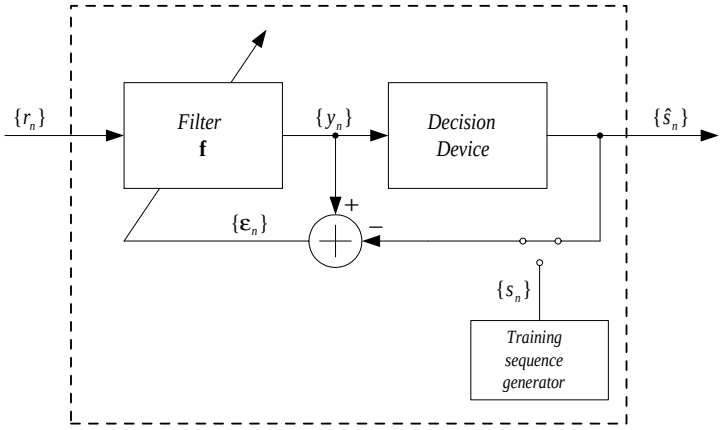

Figure 4.15: Linear equalizer operating in decision-directed (DD) mode

Chapter 5

Decision Feedback Equalizer

The theory behind the adaptive non-linear decision feedback equalizer (DFE) is introduced in this chapter together with algorithms that imply both training signals and blind updates.

5.1 Decision Feedback Equalizer Structure

The non-linear feature of the decision feedback equalizer (DFE) is due to the non-linear decision device. Figure 5.1 shows how the decisions are fed back in a feedback path to equalize a dispersive channel. The DFE is based

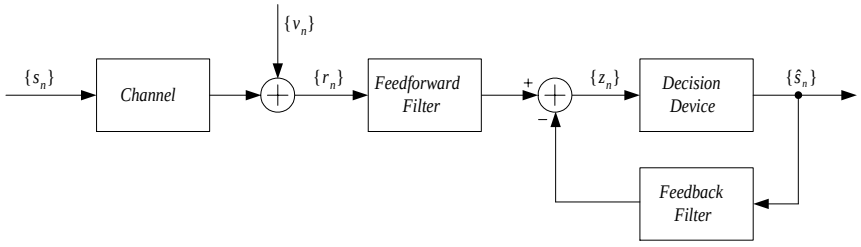

Figure 5.1: The decision feedback equalizer

on the principle that once a transmitted symbol has been estimated, the ISI contribution of that symbol to future received symbols can be exactly removed. Thus, when a symbol has been estimated, the filter structure can compute the ISI effect it would tend to have on subsequent received symbols and compensate the input to the decision device for the next samples. This *postcursor* ISI removal is accomplished by the use of a feedback filter structure. The filter in the feedback path has, as its input, the output of the decision device (the hard symbol estimates). The tap-weights converge through

a LMS-like process to resemble the tail of the channel impulse response (taps beyond the center tap). While the linear FIR equalizer only have zeros to use when equalizing a fading channel that usually itself consist of only zeros, the DFE can by using its feedback feature better eliminate the ISI. This is specially the case if the channel zeros are located close to the unit circle. For example to equalize the channel $c(q) = 1 + 0.9q^{-1}$ using a linear equalizer, the inverse filter $f(q) = 1/c(q) = 1 - 0.9q^{-1} + 0.9^2q^{-2} - 0.9^3q^{-3} + \dots$ is needed. It is here the advantage of the DFE comes into place, these "mean" zeros can be put in feedback path where no inversion is needed and thereby is the need of an "extra" long FIR filter eliminated.

Figure 5.5 shows the FIR DFE in more detail and for a more thorough presentation of the DFE we first consider the un-equalized channel response of Figure 5.2. This response is of course obtained at the output of the channel, prior the equalizer. The purpose of the feedforward filter in a DFE is

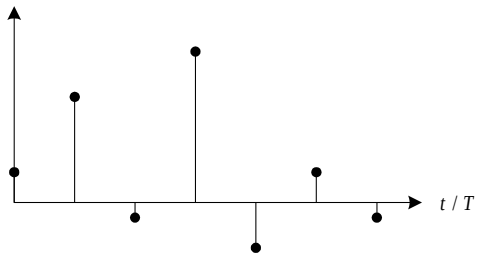

Figure 5.2: The channel impulse response

to suppress the first $\delta - 1$ taps of the channel response (the *precursor* ISI), and to keep tap δ (the *reference* or *cursor* tap) close to unity. Figure 5.3 shows the response at the output of the feedforward filter. The *postcursor*

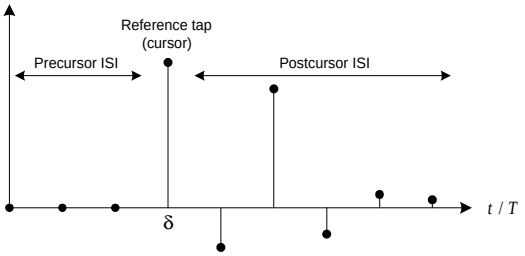

Figure 5.3: The partly equalized impulse response at the output of the feedforward filter

ISI consist of the arbitrary taps $\delta + 1$, $\delta + 2$, etc. and it is the feedback part's task to suppress these taps. When the output of the feedback filter is subtracted from the output of the feedforward filter, the equalized impulse

response of Figure 5.4 is obtained. Note how both pre- and postcursor ISI has been removed, resulting in a equalized response in the form of a pure delay δ, generating zero ISI. The performance of the DFE depends strongly

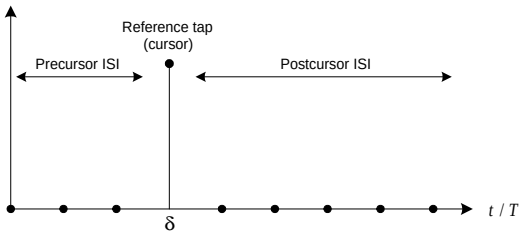

Figure 5.4: The final equalized response

on the accuracy of the estimated and fed back decisions. As a decrease in SNR will give us more unreliable estimates, the DFE will not perform too well in low SNR environments. Simulations show that the trained DFE has superior performance compared to that of the trained linear equalizer, especially if the channel experiences strong fading and noise. Another feature of the DFE is that it does not introduce any noise enhancement in the feedback path, since symbols that are impaired by noise are passed onto a decision device before they are fed back. The decision device quantize the symbols into hard decisions and eliminates thereby any noise. This reasoning is of course only valid if correct hard estimates are assumed.

When obtaining the total channel-equalizer impulse response, we have to assume *correct past decisions* (CPD). The reason for doing this is that we want to by-pass the non-linear decision device. If we assume CPD, i.e. $\hat{s}_n = s_{n-\delta} = q^{-\delta}s_n$, where δ is an unavoidable (integer) delay that is introduced by the channel and feedforward filter combination. From Figure 5.1 it is then evident that we can express the equalized soft output as $z_n = [c(q)f(q) - q^{-\delta}g(q)]s_n + f(q)v_n = h^{\text{tot}}(q)s_n + f(q)v_n$, where $f(q) = f_0 + f_1q^{-1} + \ldots + f_{L_f-1}q^{-L_f+1}$ represents the feedforward filter and $g(q) = g_1q^{-1} + \ldots + g_{L_g}q^{-L_g}$ the feedback filter ($c(q)$ represents the channel as previously). Hence, in the assumption of CPD and absence of noise, the total system impulse response becomes $h^{\text{tot}}(q) = c(q)f(q) - q^{-\delta}g(q)$.

5.2 Trained Algorithms

5.2.1 The Least Mean Square DFE

The LMS-DFE is also derived by minimizing the MSE, i.e. the LMS algorithm that was derived in Section 4.2.2 can also be used for the adaptation of both the feedforward and feedback filters of the DFE [15].

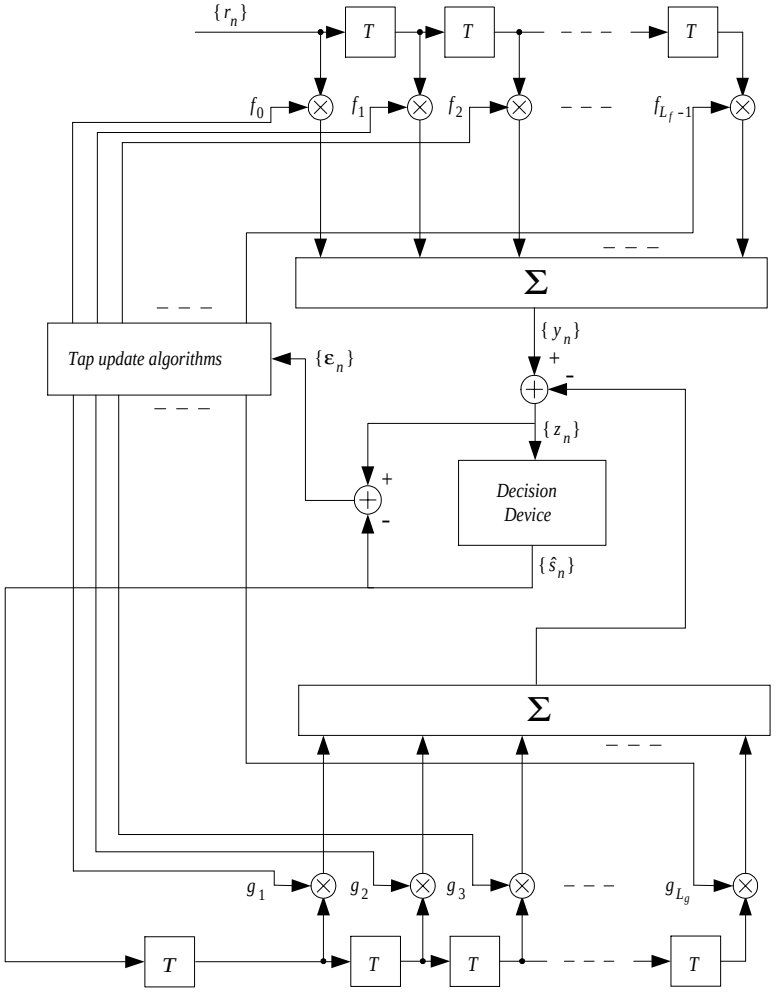

Figure 5.5: Decision feedback equalizer in decision directed mode implemented as two FIR filters of length L_f and L_g, respectively

We define the feedforward filter tap-weight vector:

$$\mathbf{f} \triangleq [f_0 \quad f_1 \quad f_2 \quad \cdots \quad f_{L_f-1}]^T \tag{5.1}$$

and the feedback filter's equivalent:

$$\mathbf{g} \triangleq [-g_1 \quad -g_2 \quad -g_3 \quad \cdots \quad -g_{L_g}]^T \tag{5.2}$$

The regressor for the feedforward filter is the same as for the linear equalizer:

$$\mathbf{r}_n = \mathbf{r}[n] \triangleq [r_n \quad r_{n-1} \quad r_{n-2} \quad \cdots \quad r_{n-L_f+1}]^T \tag{5.3}$$

and we define the feedback filter's regressor of training symbols as:

$$\mathbf{s}_{n-\delta} = \mathbf{s}[n-\delta] \triangleq [s_{n-\delta-1} \quad s_{n-\delta-2} \quad s_{n-\delta-3} \quad \cdots \quad s_{n-\delta-L_g}]^T \tag{5.4}$$

As previously mentioned, when adapting the feedforward and feedback filter taps, we can, as in the case for the linear equalizer minimize the MSE. Therefore we will have an adaptation algorithm for the feedback filter coefficients similar to that of the feedforward filter. Thus, it is convenient to define the total DFE tap-weight vector and total DFE regressor as:

$$\begin{aligned} \mathbf{\Theta} &\triangleq [f_0 \quad \cdots \quad f_{L_f-1} \quad -g_1 \quad \cdots \quad -g_{L_g}]^T \\ &= \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} \end{aligned} \tag{5.5}$$

$$\begin{aligned} \mathbf{\Phi}_n = \mathbf{\Phi}[n] &\triangleq [r_n \quad \cdots \quad r_{n-L_{f+1}} \quad s_{n-\delta-1} \quad \cdots \quad s_{n-\delta-L_g}]^T \\ &= \begin{bmatrix} \mathbf{r}_n \\ \mathbf{s}_{n-\delta} \end{bmatrix} \end{aligned} \tag{5.6}$$

Now, we can generate the soft estimate as the inner-product $z_n = \mathbf{\Theta}^T \mathbf{\Phi}[n]$. These soft estimates $\{z_n\}$ are the input to the decision device in Figure 5.1, and is the resulting signal when subtracting the output of the feedback filter from the output of the feedforward filter.

The LMS-DFE algorithm adaptively minimizes the MSE cost:

$$\begin{aligned} J^{\text{mse}} &\triangleq \frac{1}{2}\mathrm{E}\{|\varepsilon_n|^2\} \\ &= \frac{1}{2}\mathrm{E}\{|z_n - s_{n-\delta}|^2\} \\ &= \frac{1}{2}\mathrm{E}\{|\mathbf{\Theta}^T\mathbf{\Phi}[n] - s_{n-\delta}|^2\} \end{aligned} \tag{5.7}$$

and can be derived in a similar fashion as for the linear equalizer. Thus the resulting LMS-DFE algorithm is without any derivation presented as:

$$\begin{aligned} \mathbf{\Theta}[n+1] &= \mathbf{\Theta}[n] - \mu\varepsilon_n\mathbf{\Phi}^\star[n] \\ &= \mathbf{\Theta}[n] - \mu(z_n - s_{n-\delta})\mathbf{\Phi}^\star[n] \\ &= \mathbf{\Theta}[n] - \mu(\mathbf{\Theta}^T[n]\mathbf{\Phi}[n] - s_{n-\delta})\mathbf{\Phi}^\star[n] \end{aligned} \tag{5.8}$$

Figure 5.6 shows SER histories for a trained linear equalizer using 1,000 initial training symbols and its decision feedback equivalent, and Figures 5.7 and 5.8 show the equalized constellations after 500,000 iterations. Note the enormous difference in performance between the linear and decision feedback equalizers (especially for SNR = 20dB). The trained DFE is clearly advantageous compared to its linear counterpart and perfectly equalizes strongly fading (notch depths of 30 dB and more) channels in high SNRs.

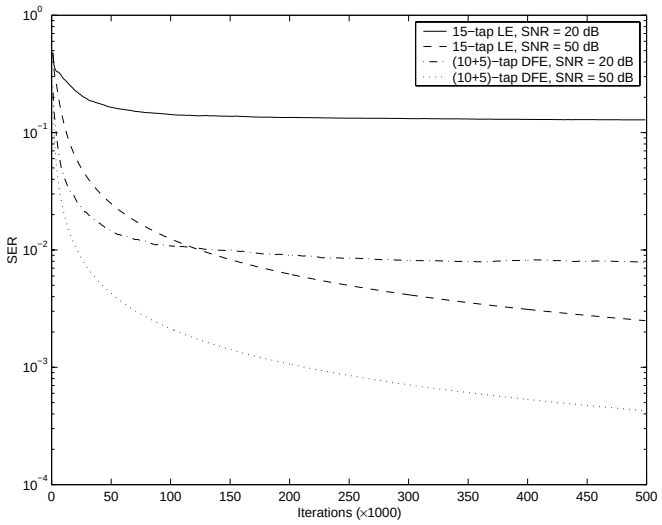

Figure 5.6: Example of the SER history for trained linear and decision feedback equalizers and different SNRs. The channel is strongly fading with additive noise

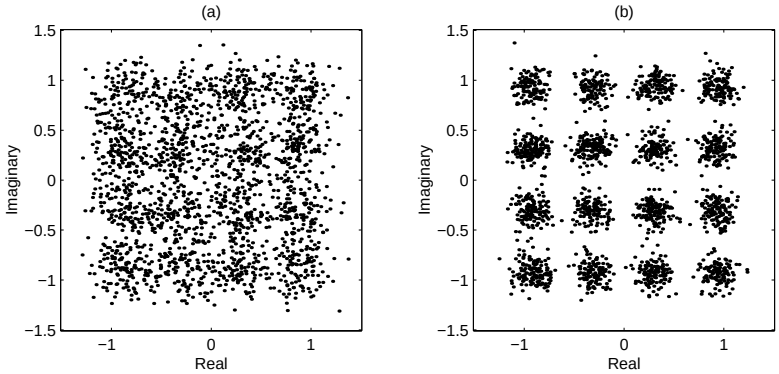

Figure 5.7: Last 2,000 symbols of the equalized constellations for (a) the linear, and (b) decision feedback equalizer. The channel is that of Figure 5.6 and 20 dB SNR

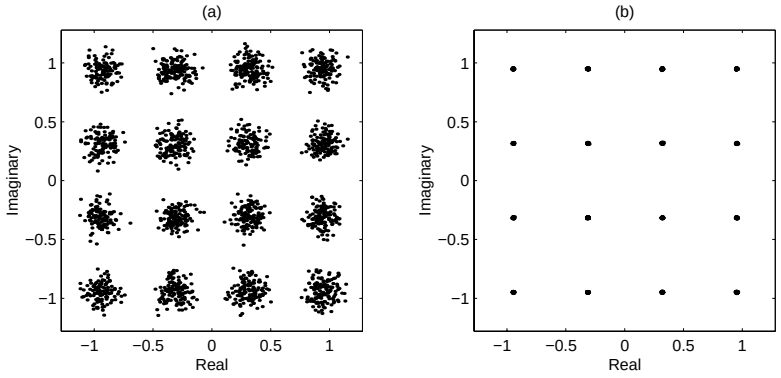

Figure 5.8: Last 2,000 symbols of the equalized constellations for (a) the linear, and (b) decision feedback equalizer (note the perfect constellation!). The channel is that of Figure 5.6 and 50 dB SNR

5.3 Blind Algorithms

As we noticed for the LMS-DFE, we ended up with similar tap-update algorithms as for the linear equalizer. This is the case when gradient descent methods are used, so less surprisingly we will also here end up with blind algorithms that resemble the ones for the blind linear equalizer.

5.3.1 The Constant Modulus DFE

Without any further motivation we here define the CM cost function:

$$\begin{aligned} J^{\mathrm{cm}} &\triangleq \frac{1}{4}\mathrm{E}\{(|z_n|^2-\gamma)^2\} \\ &= \frac{1}{4}\mathrm{E}\{(|\boldsymbol{\Theta}^T\hat{\boldsymbol{\Phi}}[n]|^2-\frac{\mathrm{E}\{|s_n|^4\}}{\sigma_s^2})^2\} \end{aligned} \tag{5.9}$$

where $\boldsymbol{\Theta}$ is previously defined by (5.5) and the regressor $\hat{\boldsymbol{\Phi}}[n]$ of received sampled symbols and previously quantized symbols is defined as:

$$\begin{aligned} \hat{\boldsymbol{\Phi}}_n = \hat{\boldsymbol{\Phi}}[n] &\triangleq [r_n \quad \cdots \quad r_{n-L_f+1} \quad \hat{s}_{n-1} \quad \cdots \quad \hat{s}_{n-L_g}]^T \\ &= \begin{bmatrix} \mathbf{r}_n \\ \hat{\mathbf{s}}_n \end{bmatrix} \end{aligned} \tag{5.10}$$

where $\{\hat{s}_n\}$ are the quantized and fed back symbol estimates.

The following tap-update algorithm, named CMA-DFE, iteratively minimizes the CM cost function:

$$\begin{aligned}\boldsymbol{\Theta}[n+1] &= \boldsymbol{\Theta}[n] - \mu\varepsilon_n^{\text{cma}}\hat{\boldsymbol{\Phi}}^{\star}[n] \\ &= \boldsymbol{\Theta}[n] - \mu z_n(|z_n|^2 - \gamma)\hat{\boldsymbol{\Phi}}^{\star}[n] \\ &= \boldsymbol{\Theta}[n] - \mu z_n(|\boldsymbol{\Theta}^T[n]\hat{\boldsymbol{\Phi}}[n]|^2 - \gamma)\hat{\boldsymbol{\Phi}}^{\star}[n] \end{aligned} \tag{5.11}$$

The feedforward filter can be initialized with a normalized single-spike somewhere in the center of the tapped-delay line and the feedback filter is initialized to zero. This initial setting corresponds to a pure delay and does not perform any form of equalization.

Little can be found in literature regarding blind DFEs sensitivity to decision errors and convergence properties etc. This is mainly due to its stochastic and non-linear nature, and therefore making it hard to study. Although, [3] investigates blind DFE's more closely, and simulations in this thesis show some improvement for CMA-DFE compared to a linear equalizer updated using CMA.

5.3.2 Fractionally Spaced CMA-DFE

The feedforward filter of the DFE can be fractionally-spaced (FS) as in Section 4.3.2, but the feedback filter is baud-spaced since it has the fed back baud-spaced decision as its input. Thus, the overall BS impulse response for a two-times FS channel and equalizer combination can be obtained by assuming CPD and no noise as $h^{\text{tot}}(q) = [c(\tilde{q})f(\tilde{q})]_{\downarrow 2} - q^{-\delta}g(q)$, where q^{-1} and $\tilde{q}^{-1}$ are the BS unit delay and the two-times FS unit delay operators respectively.

The two-times FS-CMA-DFE can be updated as follows:

$$\begin{aligned}\boldsymbol{\Theta}[n+1] &= \boldsymbol{\Theta}[n] - \mu\varepsilon_n^{\text{cma}}\hat{\boldsymbol{\Phi}}^{\star}[n] \\ &= \boldsymbol{\Theta}[n] - \mu z_n(|z_n|^2 - \gamma)\hat{\boldsymbol{\Phi}}^{\star}[n] \\ &= \boldsymbol{\Theta}[n] - \mu z_n(|\boldsymbol{\Theta}^T[n]\hat{\boldsymbol{\Phi}}[n]|^2 - \gamma)\hat{\boldsymbol{\Phi}}^{\star}[n] \end{aligned} \tag{5.12}$$

where the tap vector is two-times fractionally-spaced in the feedforward section and baud-spaced in the feedback section, and the both FS/BS regressor vector is at baud instant n defined as:

$$\hat{\boldsymbol{\Phi}}_n = \hat{\boldsymbol{\Phi}}[n] \triangleq [r_{2n} \quad \cdots \quad r_{2n-L_f+1} \quad \hat{s}_{n-1} \quad \cdots \quad \hat{s}_{n-L_g}]^T \tag{5.13}$$

Here the tap-weight vector can also be initialized with a normalized single- or double-spike, somewhere in the center of the tapped-delay line.

5.3.3 Adaptive Complex Gain Control

As for the blind linear equalizer, we also need some ACGC algorithms for the CMA-DFE. The same algorithms that were presented in Section 4.3.3 can also be used for the blind DFE, and therefore does not need any further presentation.

The attenuated and phase-shifted soft decision:

$$\tilde{z}_n = G_n \exp(j\phi_n) z_n = G_n \exp(j\phi_n) \boldsymbol{\Theta}^T \hat{\boldsymbol{\Phi}}[n], \tag{5.14}$$

the gain recovery algorithm:

$$G[n+1] = G[n] + \lambda(\sigma_s^2 - |\tilde{z}_n|^2), \tag{5.15}$$

the phase-error function:

$$\varepsilon_n^{\phi} = \mathrm{sgn}(\Re\{\varepsilon_n^{\mathrm{dd}}\})\mathrm{sgn}(\Im\{\tilde{z}_n\}) - \mathrm{sgn}(\Im\{\varepsilon_n^{\mathrm{dd}}\})\mathrm{sgn}(\Re\{\tilde{z}_n\}) \tag{5.16}$$

where $\varepsilon_n^{\mathrm{dd}} = \tilde{z}_n - \hat{s}_n$,

the phase recovery loop:

$$\phi[n+1] = \phi[n] + \alpha\varepsilon_n^{\phi}, \tag{5.17}$$

and finally, the modified tap-update algorithm:

$$\boldsymbol{\Theta}[n+1] = \boldsymbol{\Theta}[n] - \mu\varepsilon_n \hat{\boldsymbol{\Phi}}^{\star}[n] G_n \exp(-j\phi_n). \tag{5.18}$$

The Figures 5.9 and 5.10 show how the feedforward and feedback filter taps of a blind CMA-DFE converge. The PLL is turned on to recover the phase after 15,000 iterations.

5.3.4 Decision-Directed Blind Adaptive DFE

After initial convergence using a training signal or some blind scheme, we can transfer the DFE's tap-update to a decision-directed mode, similar to what was done in Section 4.3.4. In DD mode the instantaneous error of the DFE is given by $\varepsilon_n^{\mathrm{dd}} = z_n - \hat{s}_n$ and by replacing the error we had in training mode or the CMA error in the blind update with $\varepsilon_n^{\mathrm{dd}}$, we receive the DD DFE update algorithm:

$$\boldsymbol{\Theta}[n+1] = \boldsymbol{\Theta}[n] - \mu\varepsilon_n^{\mathrm{dd}} \hat{\boldsymbol{\Phi}}^{\star}[n] \tag{5.19}$$

This way the equalizer can continue to minimize the ISI and adapt its parameters to "track" slow channel variations. A DFE that has been initialized with a training sequence shows superior performance in DD mode compared to a likewise initialized DD linear equalizer. This indicates that a DFE, after successful initial convergence to a "good" setting and which is transferred to DD mode will perform really well. Figure 5.11 shows a DFE operating in DD mode and Figure 6.1 of Section 6.1 shows the ISI trajectory for blind CMA-DFE that has been transferred to DD mode.

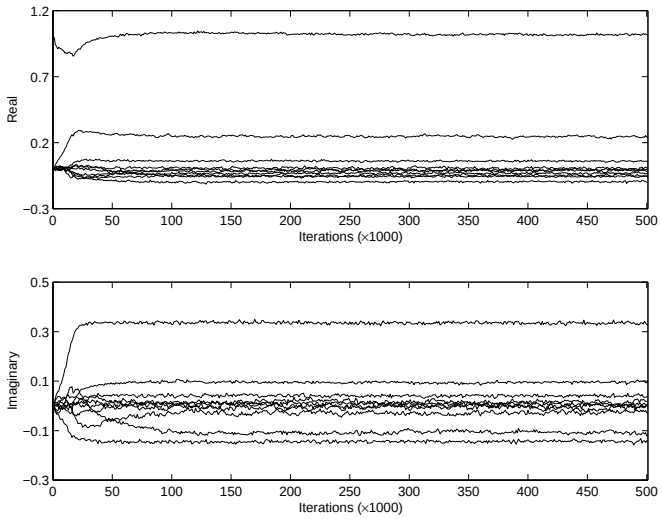

Figure 5.9: Real part (top) and imaginary part (bottom) of the tap history of the feedforward filter of a blind CMA-DFE. The PLL is turned on after 15,000 iterations and the channel is strongly fading with additive noise

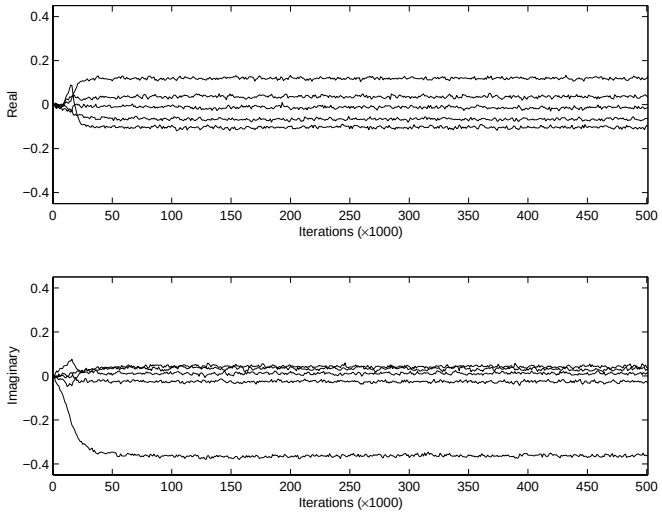

Figure 5.10: Real part (top) and imaginary part (bottom) of the tap history of the feedback filter of a blind CMA-DFE. The PLL is turned on after 15,000 iterations and the channel is strongly fading with additive noise

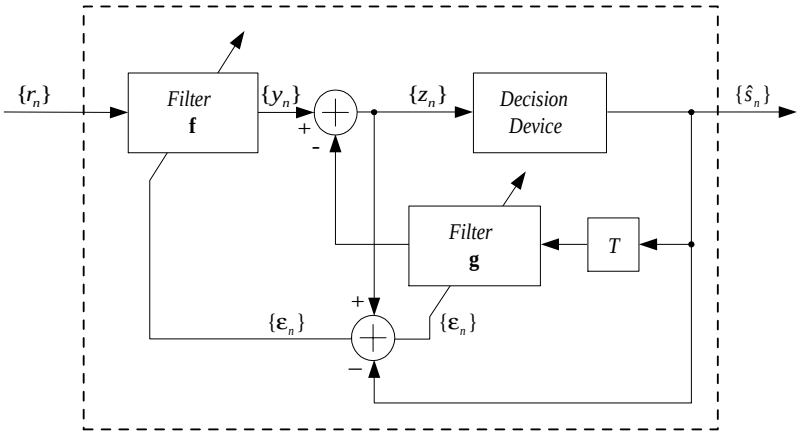

Figure 5.11: Decision-directed decision feedback equalizer (DD-DFE)

Chapter 6

Simulations

This chapter introduces some performance quantities, simulates and compares the various and previously introduced equalizers abilities to equalize fading and noisy microwave radio channels.

6.1 Performance Quantities

We need some quantities to evaluate the performance of the various equalizers. Typical performance quantities in communications are: *bit error rate* (BER) or the closely related *symbol error rate* (SER), *mean square error* (MSE), *inter-symbol interference* (ISI), and *signal-to-noise ratio* (SNR).

The symbol error rate is defined as:

$$\text{SER} \triangleq \frac{\text{number of non-zero } (s_n - \hat{s}_n)}{\text{number of symbols}} \tag{6.1}$$

where $\{s_n\}$ are the transmitted source symbols and $\{\hat{s}_n\}$ their hard estimates. When the source is differentially encoded, the SER computations should discard trivial phase errors since only the phase transitions are used for symbol detection, and therefore is absolute knowledge of the phase not required, i.e. $\hat{s}_n = e^{j\frac{\pi}{2}m} s_{n-\delta}$ for fixed $m \in \{0, 1, 2, 3\}$ for all n would form acceptable output for a differentially encoded QAM source [8].

The mean square error is defined as:

$$\text{MSE} \triangleq \frac{\sum_n |\varepsilon_n|^2}{\text{number of symbols}} \tag{6.2}$$

where the instantaneous error in training of a linear equalizer is $\varepsilon_n = y_n - s_{n-\delta}$, and for linear equalization in CMA or DD mode are the decision errors $\varepsilon_n = y_n - \hat{s}_n$ or if applicable $\varepsilon_n = \tilde{y}_n - \hat{s}_n$ used to give a measurable

performance quantity of the average dispersion of the equalized symbols away from their hard decisions, i.e. it is a measurement of how compact the equalized constellation "clouds" are. For a DFE is the soft decisions y_n or $\tilde{y}_n$ of the linear equalizer replaced by the DFE's soft decisions z_n or $\tilde{z}_n$ respectively.

A quantitative measure of the inter-symbol interference is [9]:

$$\text{ISI} \triangleq \frac{\sum_n |h_n^{\text{tot}}| - \max_n |h_n^{\text{tot}}|}{\max_n |h_n^{\text{tot}}|} \tag{6.3}$$

where $\{h_n^{\text{tot}}\}$ is the impulse response of the total system, i.e. the channel and equalizer combination, and (if applicable) it is also extended to include the ACGC. Although, note that the ACGC's multiplication with a complex scalar will not affect the ISI quantity. The ISI quantity is also known as the *peak distortion* [9]. Worth noting is that when ISI $< \frac{1}{M-1}$ for an M-PAM source, then the total system is considered to be tuned to an "open-eye" setting, and the source can (in the absence of noise) be reconstructed without any errors [9]. See Figure 6.1 for how a blind CMA-DFE that is transferred to DD mode minimizes the excess ISI compared to the non-transferred CMA-DFE.

These three defined quantities (SER, MSE, ISI) together with the signal-to-noise ratio (SNR) that was previously defined of (2.4), are frequently used in the performance simulations.

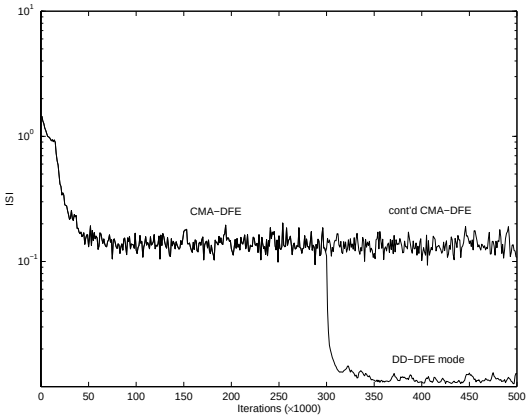

Figure 6.1: ISI trajectories for a blind CMA-DFE, and a blind CMA-DFE that is transferred to DD mode after 300,000 iterations. The channel is strongly fading with additive noise

6.2 Simulation Setup

The simulations are carried out in baseband using the model that was developed in Section 3.3. The source symbols are equally likely drawn from a normalized 16-QAM alphabet to form a source sequence with unity variance and 500,000 symbols. The additive noise is zero-mean, Gaussian and uncorrelated, i.e. white Gaussian noise. In a real situation the noise will, together with the received signal be filtered by the receiver filter. Hence, making the noise correlated at the output of the receiver filter. When evaluating the performance of various equalizers, it is more convenient to deal with white noise when computing the error-rate performance for certain SNRs [15]. Therefore, we will assume the noise to be white, and this is satisfactory as long as we evaluate the equalizers during identical conditions. Although, the fractionally-spaced systems are simulated with a low-pass equivalent of the additive white noise, otherwise they would not be comparable to the baud-spaced systems.

The discrete-time baseband FIR channel taps are acquired by sampling the continuous-time channel impulse response in equation (3.8) at baud- or twice the baud-rate. The resulting time-discrete impulse response is also normalized to eliminate flat-fading.

The blind CMA algorithms require the ACGC algorithms to recover the gain, which speeds up the convergence, but most importantly they are needed to recover the phase. All simulations involving CMA turns on the phase-recovery loop after $15,000$ iterations. When the PLL is turned on, it rotates the constellation into its original place and keeps it there.

The trained LMS uses $1,000$ initial training symbols and is transferred to DD mode after initial training. Note that the initial training is included in the performance measurements (SER, MSE) for the trained LMS equalizer. Since the training symbols are known, i.e. will not cause any errors, they should of course not contribute to the "real" symbol error rate. But, since it is desired to compare the initial convergence for the different methods, the training errors will not be discarded. Thus, causing a final SER value for the trained LMS that is larger than the "real" SER, especially since it is during training that most errors occur. It would not be very illustrative to display SER curves that are identical to zero (which could be the case after long enough training) when investigating the initial convergence of LMS. Although, the "real" SER values, which are denoted $\mathrm{SER}_{\mathrm{dd}}$ for trained LMS, are also computed and are included in the result tables for comparison.

After initial convergence with blind CMA the equalizer update can also be switched to DD mode (CMA/DD). This switch is done after $300,000$ iterations. Hence, the last $200,000$ iterations are in these simulations carried out in DD mode.

The signal-to-noise ratio is 40 dB in all simulations, except for the ones testing the performance in different SNR environments. They have SNRs ranging from $10 - 50$ dB.

Baud-spaced (BS) equalization is always assumed if nothing else stated. Fractionally-spaced equalization is always denoted by FS.

6.3 Channels

6.3.1 Minimum Phase Channel: A

Channel Setup

- An overall raised-cosine shaped impulse response with 35% (*roll-off factor = 0.35*) excess bandwidth equally distributed between the transmitter and receiver
- 30 dB notch depth in minimum phase with a displacement of $\Delta f = 10$ MHz
- 9 BS sampled and normalized FIR channel taps, and 18 FS sampled and normalized FIR channel taps

Two BS and one FS discrete-time versions of this channel are used. The BS versions differ a half symbol period in sampling-phase. The two BS channels and the FS channel are denoted *Channel-A*, *Channel-AA*, and *FS-Channel-A*, respectively. Figures 6.2 and 6.3 show the sampled impulse response, channel zeros, and the frequency response for Channel-A, and Channel-AA, respectively.

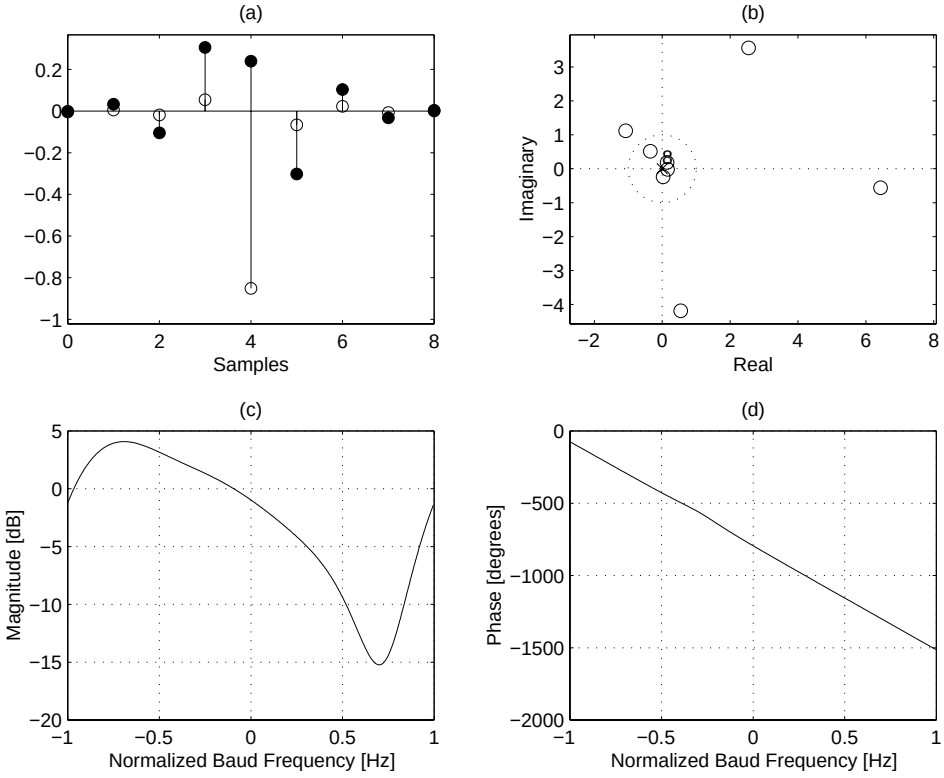

Figure 6.2: Discrete Channel-A characteristics: (a) impulse response (real/imaginary $\Leftrightarrow$ filled/non-filled), (b) channel zeros, (c) magnitude response, (d) phase response

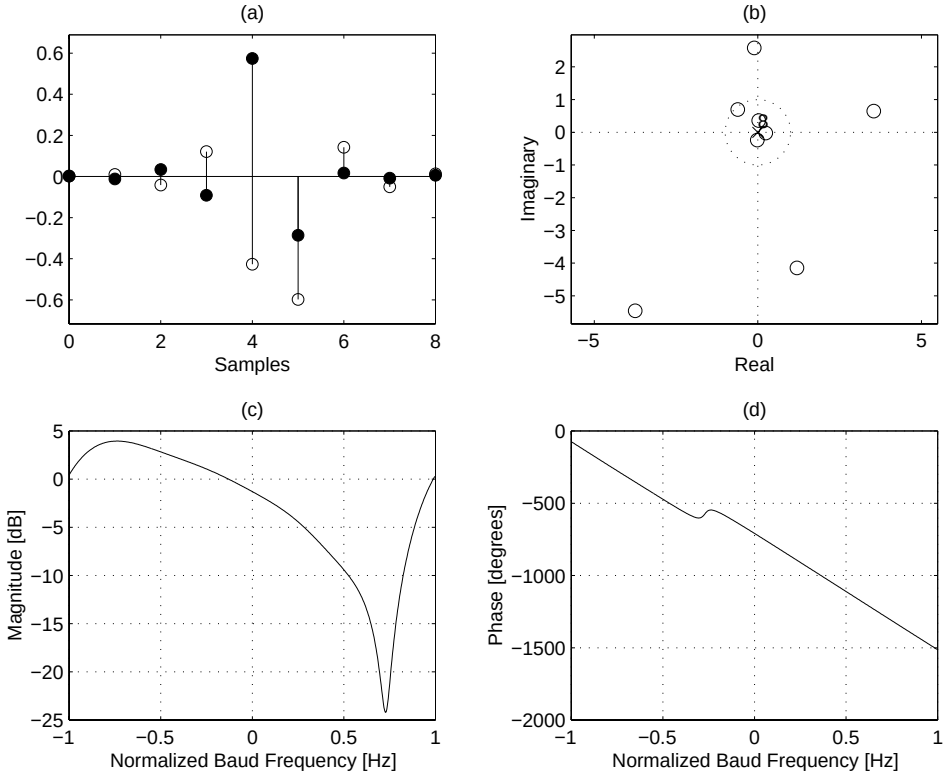

Figure 6.3: Discrete Channel-AA characteristics: (a) impulse response (real/imaginary $\Leftrightarrow$ filled/non-filled), (b) channel zeros, (c) magnitude response, (d) phase response

6.3.2 Non-minimum Phase Channel: B

Channel Setup

- An overall raised-cosine shaped impulse response with 35% (*roll-off factor = 0.35*) excess bandwidth equally distributed between the transmitter and receiver
- 30 dB notch depth in non-minimum phase with a displacement of $\Delta f = 5$ MHz
- 9 BS sampled and normalized FIR channel taps, and 18 FS sampled and normalized FIR channel taps

One BS and one FS discrete-time versions of this channel are used. The BS, and the FS channels are denoted *Channel-B*, and *FS-Channel-B*, respectively. Figure 6.4 shows the sampled impulse response, channel zeros, and the frequency response for Channel-B.

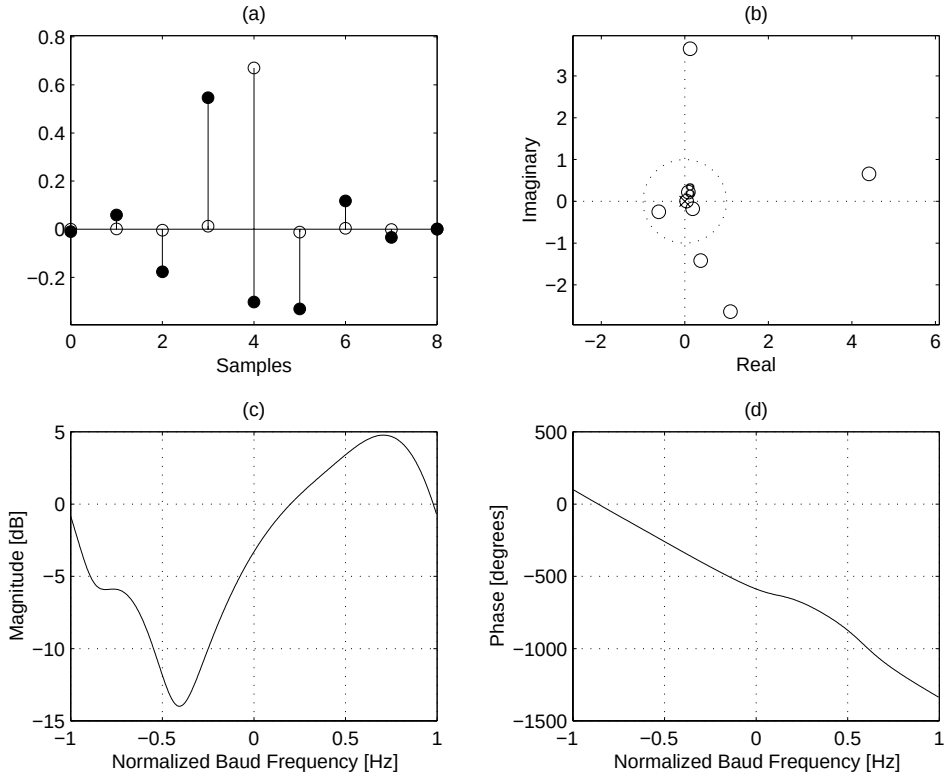

Figure 6.4: Discrete Channel-B characteristics: (a) impulse response (real/imaginary ⇔ filled/non-filled), (b) channel zeros, (c) magnitude response, (d) phase response

Variable	Trained LMS	BS-CMA	FS-CMA
L_f	15	15	30
$\mathbf{f}[0]$	single center spike	single center spike	double center spike
μ_0	1×10^{-2}	5×10^{-4}	5×10^{-4}
μ_{dd}	5×10^{-3}	n/a	n/a
λ	n/a	1×10^{-4}	1×10^{-4}
α	n/a	1×10^{-5}	1×10^{-5}
G_0	n/a	1	1
ϕ_0	n/a	$\pi/4$	$\pi/4$

Table 6.1: Linear equalizer setup

Variable	BS-CMA/DD	FS-CMA/DD
L_f	15	30
$\mathbf{f}[0]$	single center spike	double center spike
μ_0	5×10^{-4}	5×10^{-4}
μ_{dd}	5×10^{-4}	5×10^{-4}
λ	1×10^{-4}	1×10^{-4}
α	1×10^{-5}	1×10^{-5}
G_0	1	1
ϕ_0	$\pi/4$	$\pi/4$

Table 6.2: Continued linear equalizer setup

6.4 Linear Equalizer Simulation

The used linear equalizers are: trained LMS, blind BS-CMA, blind two-times FS-CMA, and their DD versions.

Equalizer Setup

Tables 6.1 and 6.2 summarize the linear equalizer setups.

Simulation Results

Figure 6.5 shows SER curves for various linear equalizers operating on Channel-A and FS-Channel-A. The SER curves for the equalizers operating in DD mode are identical to the ones in non-DD mode. The reason being is that the update is transferred to DD-mode after that the initial convergence does not produce any more symbol errors, i.e. the trajectories continue to follow each other if no more errors are produced. Figure 6.6 shows SER curves for the linear equalizers operating on Channel-B and FS-Channel-B. While the SER curves do not show any performance difference for transition

Quantity	Trained BS-LMS	BS-CMA	FS-CMA
SER_{tot}	1.1×10^{-3}	7.0×10^{-2}	5.4×10^{-2}
SER_{dd}	4.0×10^{-4}	n/a	n/a
$SER_{300k-500k}$	n/a	0	0
MSE_{tot}	2.1×10^{-3}	1.3×10^{-2}	1.3×10^{-2}
MSE_{dd}	1.7×10^{-3}	n/a	n/a
$MSE_{300k-500k}$	n/a	3.8×10^{-3}	4.1×10^{-3}
ISI	1.2×10^{-1}	2.4×10^{-1}	2.2×10^{-1}

Table 6.3: Performance results for linear equalization of Channel-A and FS-Channel-A, SNR = 40 dB

from blind CMA to blind DD mode, the smoothed instantaneous squared error histories of Figures 6.7 and 6.8 show the increased error performance for blind linear equalization of Channel-A, Channel-B, and their FS versions respectively. The update is transferred to DD mode after 300,000 iterations.

Figure 6.9 shows how the linear equalizers perform in different SNR environments of Channel-AA and FS-Channel-A. The SER values are the final ones obtained after 500,000 iterations.

Plots of responses and zeros of some channels, linear equalizers, and combinations thereof, is found in Section B.1. Tables 6.3 and 6.4 summarize the obtained performance quantities when equalizing Channel-A and FS-Channel-A.

Tables 6.5 and 6.6 summarize the obtained performance quantities when equalizing Channel-B and FS-Channel-B. Note that the quantities SER_{tot}, and MSE_{tot}, are the total values after 500,000 iterations and include the initial training for the trained LMS equalizer. The quantities SER_{dd}, and MSE_{dd}, are the values obtained in DD mode for the trained LMS, i.e. the "real" values. Furthermore, the quantities $SER_{300k-500k}$, and $MSE_{300k-500k}$, are the values obtained between iteration number $300,000 - 500,000$, which is DD mode for the BS-CMA/DD, and FS-CMA/DD equalizers. The ISI values are the ones obtained when using the final equalizer tap values after 500,000 iterations.

Quantity	BS-CMA/DD	FS-CMA/DD
$\mathrm{SER}_{\mathrm{tot}}$	7.0×10^{-2}	5.4×10^{-2}
$\mathrm{SER}_{300\mathrm{k}-500\mathrm{k}}$	0	0
$\mathrm{MSE}_{\mathrm{tot}}$	1.2×10^{-2}	1.1×10^{-2}
$\mathrm{MSE}_{300\mathrm{k}-500\mathrm{k}}$	1.3×10^{-3}	8.0×10^{-4}
ISI	8.3×10^{-2}	4.0×10^{-2}

Table 6.4: Continued performance results for linear equalization of Channel-A and FS-Channel-A, SNR = 40 dB

Quantity	Trained BS-LMS	BS-CMA	FS-CMA
$\mathrm{SER}_{\mathrm{tot}}$	1.6×10^{-3}	3.4×10^{-2}	3.1×10^{-2}
$\mathrm{SER}_{\mathrm{dd}}$	5.0×10^{-4}	n/a	n/a
$\mathrm{SER}_{300\mathrm{k}-500\mathrm{k}}$	n/a	0	0
$\mathrm{MSE}_{\mathrm{tot}}$	6.0×10^{-3}	1.3×10^{-2}	9.3×10^{-3}
$\mathrm{MSE}_{\mathrm{dd}}$	5.3×10^{-3}	n/a	n/a
$\mathrm{MSE}_{300\mathrm{k}-500\mathrm{k}}$	n/a	8.7×10^{-3}	4.6×10^{-3}
ISI	2.1×10^{-1}	3.2×10^{-1}	2.1×10^{-1}

Table 6.5: Performance results for linear equalization of Channel-B and FS-Channel-B, SNR = 40 dB

Quantity	BS-CMA/DD	FS-CMA/DD
$\mathrm{SER}_{\mathrm{tot}}$	3.4×10^{-2}	3.1×10^{-2}
$\mathrm{SER}_{300\mathrm{k}-500\mathrm{k}}$	0	0
$\mathrm{MSE}_{\mathrm{tot}}$	1.1×10^{-2}	8.1×10^{-3}
$\mathrm{MSE}_{300\mathrm{k}-500\mathrm{k}}$	4.9×10^{-3}	1.4×10^{-3}
ISI	2.0×10^{-1}	8.5×10^{-2}

Table 6.6: Continued performance results for linear equalization of Channel-B and FS-Channel-B, SNR = 40 dB

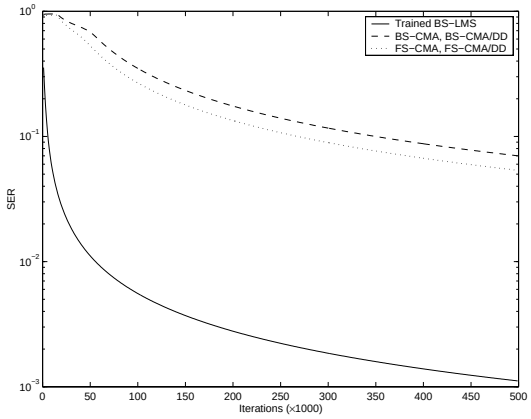

Figure 6.5: SER curves for linear equalization of Channel-A and FS-Channel-A, SNR = 40 dB

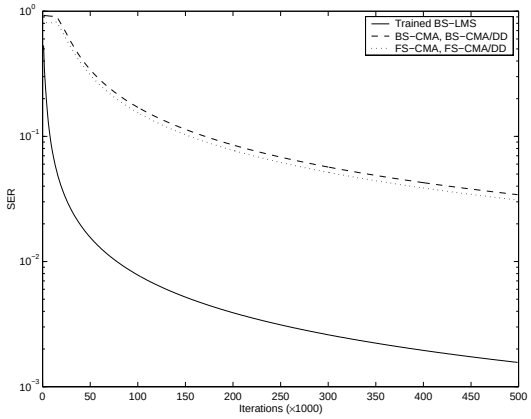

Figure 6.6: SER curves for linear equalization of Channel-B and FS-Channel-B, SNR = 40 dB

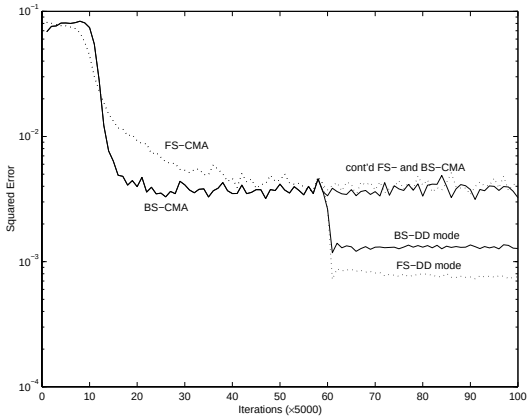

Figure 6.7: Smoothed instantaneous squared error histories for blind linear equalization of Channel-A and FS-Channel-A, SNR = 40 dB

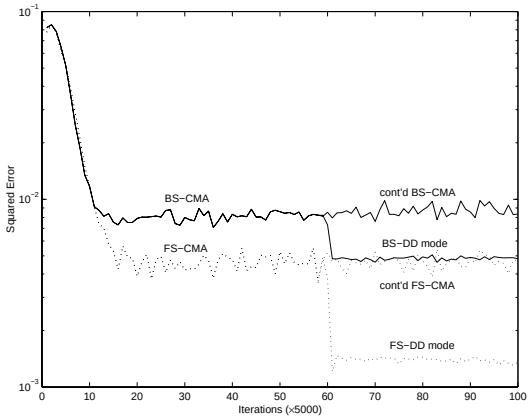

Figure 6.8: Smoothed instantaneous squared error histories for blind linear equalization of Channel-B and FS-Channel-B, SNR = 40 dB

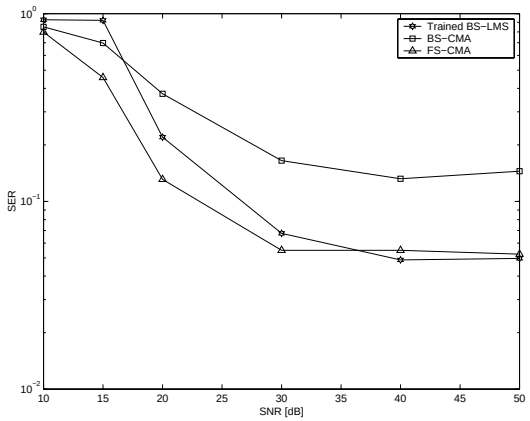

Figure 6.9: SER versus SNR after 500,000 iterations of linear equalization of Channel-AA and FS-Channel-A

Variable	Trained BS-LMS-DFE	BS-CMA-DFE	FS-CMA-DFE
L_f	10	10	20
L_g	5	5	5
$\mathbf{f}[0]$	single center spike	single center spike	double center spike
$\mathbf{g}[0]$	zero	zero	zero
μ_0	1×10^{-2}	5×10^{-4}	5×10^{-4}
μ_{dd}	5×10^{-3}	n/a	n/a
λ	n/a	1×10^{-4}	1×10^{-4}
α	n/a	1×10^{-5}	1×10^{-5}
G_0	n/a	1	1
ϕ_0	n/a	$\pi/4$	$\pi/4$

Table 6.7: Decision feedback equalizer setup

6.5 Decision Feedback Equalizer Simulation

The used decision feedback equalizers are: trained LMS-DFE, blind BS-CMA-DFE, blind two-times FS-CMA-DFE, and their DD versions.

Equalizer Setup

Tables 6.7 and 6.8 summarize the setup for the decision feedback equalizers.

Variable	BS-CMA/DD-DFE	FS-CMA/DD-DFE
L_f	10	20
L_g	5	5
$\mathbf{f}[0]$	single center spike	double center spike
$\mathbf{g}[0]$	zero	zero
μ_0	5×10^{-4}	5×10^{-4}
μ_{dd}	5×10^{-4}	5×10^{-4}
λ	1×10^{-4}	1×10^{-4}
α	1×10^{-5}	1×10^{-5}
G_0	1	1
ϕ_0	$\pi/4$	$\pi/4$

Table 6.8: Continued decision feedback equalizer setup

Quantity	Trained BS-LMS-DFE	BS-CMA-DFE	FS-CMA-DFE
SER_{tot}	2.7×10^{-4}	5.4×10^{-2}	1.3×10^{-1}
SER_{dd}	0	n/a	n/a
$\text{SER}_{300\text{k}-500\text{k}}$	n/a	0	0
MSE_{tot}	1.1×10^{-3}	1.1×10^{-2}	1.8×10^{-2}
MSE_{dd}	8.0×10^{-4}	n/a	n/a
$\text{MSE}_{300\text{k}-500\text{k}}$	n/a	5.0×10^{-3}	3.4×10^{-3}
ISI	7.1×10^{-2}	2.1×10^{-1}	1.7×10^{-1}

Table 6.9: Performance results for decision feedback equalization of Channel-A and FS-Channel-A, SNR = 40 dB

Simulation Results

Figure 6.10 shows SER curves for various decision feedback equalizers operating on Channel-A and FS-Channel-A. Figure 6.11 shows SER curves for the DFE:s operating on Channel-B and FS-Channel-B. The smoothed squared error histories of the various blind DFEs and channel combinations are shown in Figures 6.12 and 6.13.

Figure 6.14 shows how the DFE:s perform in different SNR environments of Channel-AA and FS-Channel-A. The SER values are the final ones obtained after 500,000 iterations.

Plots of responses and zeros of some channels, DFEs, and combinations thereof, is found in Section B.1. Tables 6.9 and 6.10 summarize the obtained performance quantities when equalizing Channel-A and FS-Channel-A. The notation is the same as for the linear equalizer.

Tables 6.11 and 6.12 summarize the obtained performance quantities when equalizing Channel-B and FS-Channel-B.

Quantity	BS-CMA/DD-DFE	FS-CMA/DD-DFE
$\mathrm{SER}_{\mathrm{tot}}$	5.4×10^{-2}	1.3×10^{-1}
$\mathrm{SER}_{300\mathrm{k}-500\mathrm{k}}$	0	0
$\mathrm{MSE}_{\mathrm{tot}}$	1.1×10^{-2}	1.7×10^{-2}
$\mathrm{MSE}_{300\mathrm{k}-500\mathrm{k}}$	2.7×10^{-3}	3.5×10^{-4}
ISI	1.0×10^{-1}	1.6×10^{-2}

Table 6.10: Continued performance results for decision feedback equalization of Channel-A and FS-Channel-A, SNR = 40 dB

Quantity	Trained BS-LMS-DFE	BS-CMA-DFE	FS-CMA-DFE
$\mathrm{SER}_{\mathrm{tot}}$	5.3×10^{-4}	3.4×10^{-2}	3.2×10^{-2}
$\mathrm{SER}_{\mathrm{dd}}$	0	n/a	n/a
$\mathrm{SER}_{300\mathrm{k}-500\mathrm{k}}$	n/a	0	0
$\mathrm{MSE}_{\mathrm{tot}}$	1.6×10^{-3}	1.6×10^{-2}	1.2×10^{-2}
$\mathrm{MSE}_{\mathrm{dd}}$	1.2×10^{-3}	n/a	n/a
$\mathrm{MSE}_{300\mathrm{k}-500\mathrm{k}}$	n/a	9.4×10^{-3}	7.4×10^{-3}
ISI	4.7×10^{-2}	2.6×10^{-1}	2.1×10^{-1}

Table 6.11: Performance results for decision feedback equalization of Channel-B and FS-Channel-B, SNR = 40 dB

Quantity	BS-CMA/DD-DFE	FS-CMA/DD-DFE
$\mathrm{SER}_{\mathrm{tot}}$	3.4×10^{-2}	3.2×10^{-2}
$\mathrm{SER}_{300\mathrm{k}-500\mathrm{k}}$	0	0
$\mathrm{MSE}_{\mathrm{tot}}$	1.4×10^{-2}	1.1×10^{-2}
$\mathrm{MSE}_{300\mathrm{k}-500\mathrm{k}}$	4.4×10^{-3}	3.9×10^{-3}
ISI	1.3×10^{-1}	1.4×10^{-1}

Table 6.12: Continued performance results for decision feedback equalization of Channel-B and FS-Channel-B, SNR = 40 dB

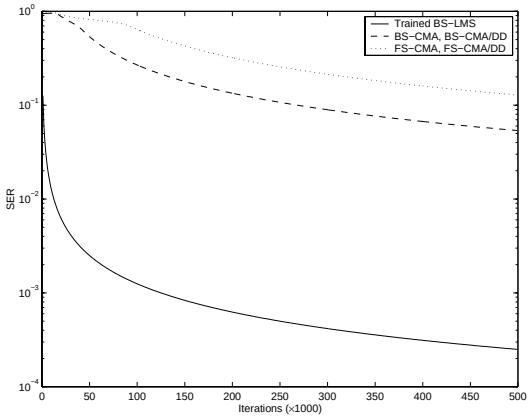

Figure 6.10: SER curves for non-linear decision feedback equalization of Channel-A and FS-Channel-A, SNR = 40 dB

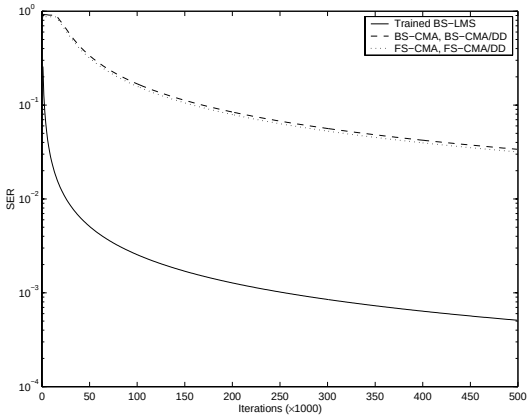

Figure 6.11: SER curves for non-linear decision feedback equalization of Channel-B and FS-Channel-B, SNR = 40 dB

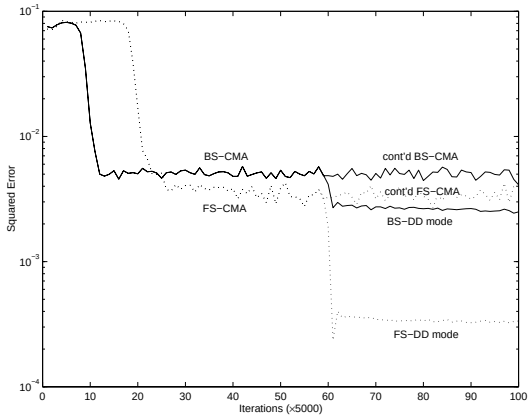

Figure 6.12: Smoothed instantaneous squared error histories for blind decision feedback equalization of Channel-A and FS-Channel-A, SNR = 40 dB

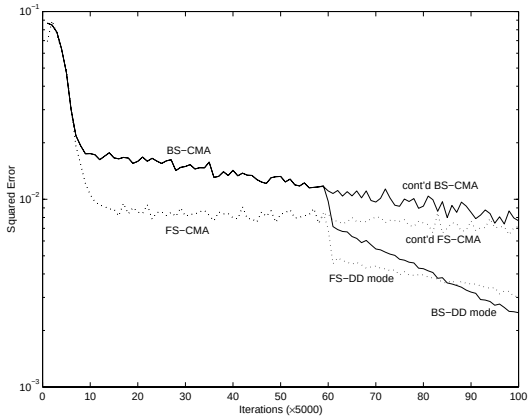

Figure 6.13: Smoothed instantaneous squared error histories for blind decision feedback equalization of Channel-B and FS-Channel-B, SNR = 40 dB

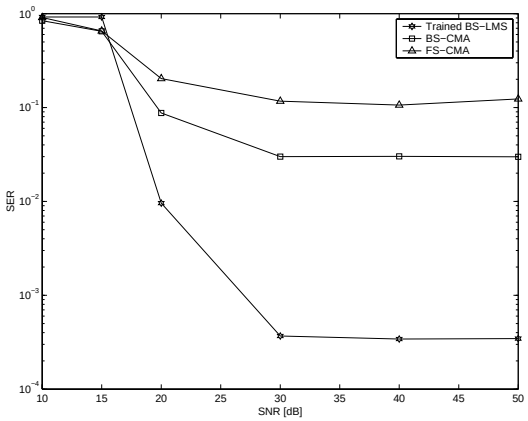

Figure 6.14: SER versus SNR after 500,000 iterations of non-linear decision feedback equalization of Channel-AA and FS-Channel-A

6.6 Blind Linear Equalizer vs Blind DFE

This section presents SER curves and squared-error history plots for blind CMA/DD-LE versus CMA/DD-DFE, and their FS versions. Figures 6.15 and 6.16 show the SER curves for blind linear and blind decision feedback equalization of Channel-A and Channel-B respectively, and Figures 6.17 and 6.18 show the instantaneous squared error histories. The equalizer setups are the same as in the previous sections. The PLL is turned on after 15,000 iterations and the CMA update is switched to DD after 300,000 iterations.

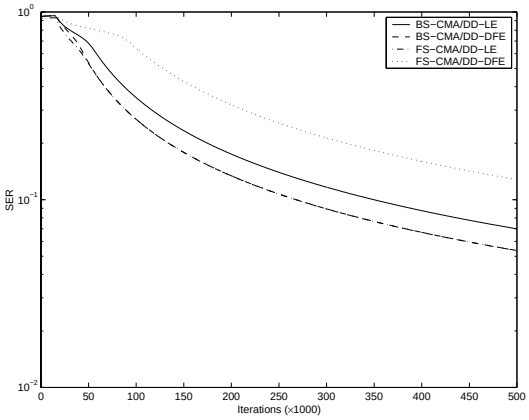

Figure 6.15: SER curves for blind LE versus blind DFE operating on Channel-A and FS-Channel-A, SNR = 40 dB

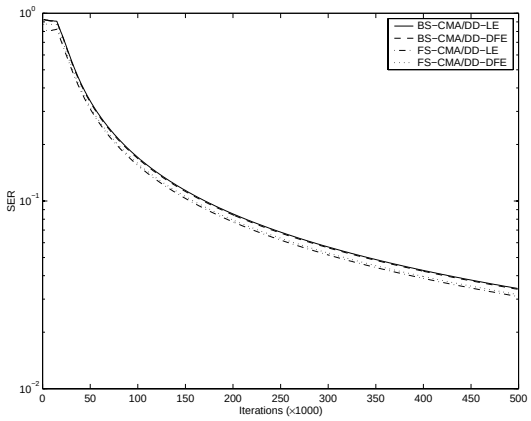

Figure 6.16: SER curves for blind LE versus blind DFE operating on Channel-B and FS-Channel-B, SNR = 40 dB

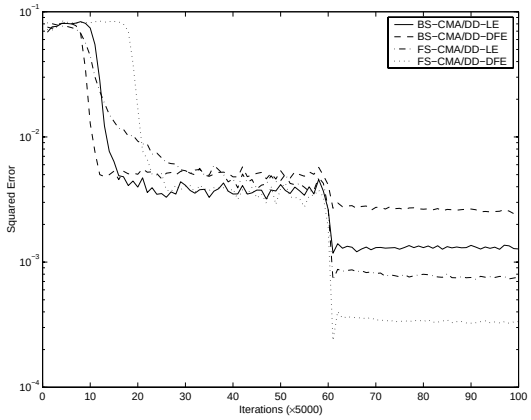

Figure 6.17: Smoothed instantaneous squared error histories for blind LE versus blind DFE operating on Channel-A and FS-Channel-A, SNR = 40 dB

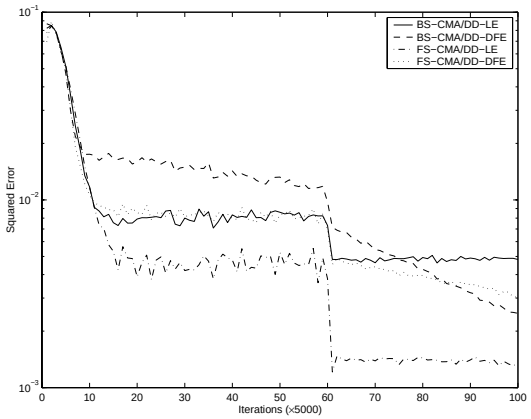

Figure 6.18: Smoothed instantaneous squared error histories for blind LE versus blind DFE operating on Channel-B and FS-Channel-B, SNR = 40 dB

Chapter 7

Conclusions

Results and conclusions drawn from the simulations of the previous chapter, and recommendations for future work are presented in this chapter.

7.1 Simulation Results and Conclusions

Firstly, it must be mentioned that it is very difficult to compare different equalizer techniques during equivalent conditions. The choice of step-sizes, initializations etc play an important roll and lead to very different results. For example, a certain initialization may be advantageous for one equalizer but disadvantageous for another equalizer operating on the same channel. Therefore the simulations must be evaluated with care and without putting too much emphasis on small performance differences between the different techniques. Although, if we do not have any knowledge of the channel conditions, the equalizers must be initialized with something general and perhaps less "tactical" and therefore the simulations may shed some light on the performance of the various techniques.

Throughout the simulations the trained LMS equalizers (both linear and DFE) gave by far the fastest convergence and lowest MSE for the different channels. They did not give the lowest ISI, the reason being is likely due to that the step-size was a bit too large allowing fast convergence but leading to a more coarse final setting. Also note that the trained DFE was in all aspects by far better than the trained linear equalizer.

The simulations also showed that a transfer to DD mode after blind convergence with CMA gave finer settings with lower MSE and ISI for all equalizer and channel combinations.

FS-CMA and its DD version showed faster or similar SER convergence, better MSE and ISI performance (especially in DD mode) than the BS-CMA and its DD version for all linear equalizers and channel combinations. The FS-CMA-DFE and its DD version gave some better and some worse performance than the BS-CMA-DFE and its DD version. A probable cause could be that the shorter length of the feedforward filter of the FS-DFE, compared to the linear FSE, does not suppress the pre-cursor ISI well enough. Thus, the improvements of the FSE were mainly for the linear FS-CMA. The expected better SER performance of FS equalization did not stand out in the simulations, but further testing of the initialization of the FSE showed that the performance of the FSE was quite sensitive to the location of the initial double-spike. When the double-spike was placed around $\lfloor \frac{L_f}{4} \rfloor$ or $\lfloor \frac{3L_f}{4} \rfloor$, instead of the center double-spike initialization (which was used in the simulations), about 2-times better SER performance (final SER) was obtained for a 20-tap linear FSE operating on Channel-A.

All blind versions of the DFE did show both better and worse performance compared to the blind linear equalization of Channel-A and Channel-B. This shows that it is hard for CMA to find a good setting when it is used for updating a DFE. Although, the SER versus SNR simulations for Channel-AA did show that blind DFEs were better than their linear versions. The reason being was that Channel-AA had a zero very close to the unit circle (see Figure 6.3) making it hard for the linear equalizer to perform well. This "bad" zero is due to poor timing when sampling and the zero also is the reason that the trained linear equalizer in Figure 6.9 performs poorly compared to the blind FS-CMA which also shows that the FSE is less sensitive to timing offsets than the BSE.

With these observations, we can draw the conclusion that the trained equalizers (especially the LMS-DFE) have superior performance, yet the question remains if it is worth the implementation. The LMS-DFE gave zero symbol error immediately after the training had ended, which in a system that runs on a symbol frequency of 25 MHz, means that zero symbol error is obtained after the equalizer has been turned on, from a cold start-up with strong fading, for only $40\,\mu$s[1]. This can be compared to the blind equalizers which all showed zero symbol error after 300,000 iterations (or less) which correspond to 12 ms.

To summarize, blind convergence of a DFE is non-trivial and shows little or no improvement. FS-CMA showed improvement mainly for the linear equalizers, and a transfer to DD mode leads to a further minimization of the excess MSE and ISI. The trained equalizers with the DFE in front of the linear equalizer performed best.

[1]This is of course only valid if it is assumed that all necessary operations fit into the time-slot of only one symbol period.

7.2 Recommendations

The introduction of an adaptive step-size would lead to a convergence to a finer setting and the equalizer would not "jitter" around the minima and therefore give a better equalizer setting with low MSE and ISI. The adaptation of the step-size should have large initial steps followed by small steps close to the minima, to finally tend to zero when the minima is reached.

When deciding the existence of a training sequence, carrier recovery and clock synchronization schemes together with equalization should also be considered. If carrier recovery or clock synchronization methods that imply training are far more efficient than their blind versions, the use of a training sequence could be motivated. The use of a training sequence also eliminates the need of a PLL that rotates the constellation into place, and if the choice falls in favor of the training sequence then the DFE is strongly recommended. Simulations and practical considerations regarding the length of an eventual training sequence are also recommended for further investigation.

If convergence speed is not that important (which may be the case for a point-to-point radio link with slow fading) and the choice falls on blind equalization, a linear FS-CMA equalizer transferred to DD mode would give sufficient performance and it would also be advantageous when considering the clock synchronization (timing) problem. If over-sampling is not desired for some reason, a baud-spaced linear equalizer based on CMA and DD is proposed. The blind DFE is difficult to update to a good setting when using CMA and DD, but the attractive qualities (which were confirmed by the trained DFE) of the DFE motivate further studies which hopefully may result in better convergence properties of blind DFEs.

Since stochastic signals are used, the simulations may vary little from one simulation to another. Therefore, if it is desired to investigate more closely fine performance differences, the quantities should be averaged over several, say 50, independent simulation runs to be able to draw better conclusions.

As a final remark, all of the proposed FIR equalizer functions are (according to Ericsson engineers) practically implementable in an ASIC or FPGA. The computational complexity, for all equalizers, is approximately in the size of $8L$ real-valued multiplications and $8L$ real-valued additions per symbol period (iteration), where L is the length[2] of the equalizer filter(s). A 16-bit binary representation should be sufficient to represent the equalizer parameters and operations thereof. Furthermore, the decision device can practically be implemented by using simple comparator logics.

[2]For a linear equalizer is $L = L_f$, and for a DFE is $L = L_f + L_g$.

...

Appendix A

Source Characteristics

A.1 Source Dispersion in M-QAM Constellations

The *source dispersion* constant γ is given by [8]:

$$\gamma = \frac{\mathrm{E}\{|s_n|^4\}}{\mathrm{E}\{|s_n|^2\}} = \frac{\mathrm{E}\{|s_n|^4\}}{\sigma_s^2} \tag{A.1}$$

where the (stochastic) source symbol $s_n \in \mathcal{A}$ and for a real constant $\Delta > 0$ is:

$$\mathcal{A} = \{(\pm\Delta \text{ or } \pm 3\Delta \text{ or } \pm 5\Delta \text{ or } \cdots) + j(\pm\Delta \text{ or } \pm 3\Delta \text{ or } \pm 5\Delta \pm \text{ or } \cdots)\}$$

the M-QAM alphabet.

Let the source symbols be uniformly distributed, i.e. $f_s(s_n) = 1/|\mathcal{A}|, \forall s_n \in \mathcal{A}$, where $|\mathcal{A}|$ is the length of the alphabet and is equal to M. Then computing the expectation value $\mathrm{E}\{|s_n|^4\}$ gives us:

$$\mathrm{E}\{|s_n|^4\} = \sum_{s_i \in \mathcal{A}} |s_i|^4 f_s(s_i) = \frac{1}{M} \sum_{s_i \in \mathcal{A}} |s_i|^4 \tag{A.2}$$

and the variance of the zero-mean source becomes:

$$\sigma_s^2 = \mathrm{E}\{|s_n|^2\} = \sum_{s_i \in \mathcal{A}} |s_i|^2 f_s(s_i) = \frac{1}{M} \sum_{s_i \in \mathcal{A}} |s_i|^2 \tag{A.3}$$

For a 16-QAM source ($M = 16$), the equations (A.2) and (A.3) evaluates to:

$$\mathrm{E}\{|s_n|^4\} = \frac{1}{16} \sum_{s_i \in \mathcal{A}} |s_i|^4 = 132\Delta^4 \tag{A.4}$$

$$\sigma_s^2 = \mathrm{E}\{|s_n|^2\} = \frac{1}{16}\sum_{s_i \in \mathcal{A}} |s_i|^2 = 10\Delta^2 \tag{A.5}$$

and thus, for a 16-QAM source, equation (A.1) evaluates to:

$$\gamma = \frac{\mathrm{E}\{|s_n|^4\}}{\sigma_s^2} = \frac{132\Delta^4}{10\Delta^2} = 13.2\Delta^2 \tag{A.6}$$

A.2 Source Kurtosis

Another commonly used definition is the *normalized source kurtosis* κ_s, and it is defined as [8]:

$$\kappa_s \triangleq \frac{\mathrm{E}\{|s_n|^4\}}{(\mathrm{E}\{|s_n|^2\})^2} = \frac{\mathrm{E}\{|s_n|^4\}}{\sigma_s^4} \tag{A.7}$$

Note that $\gamma = \sigma_s^2 \kappa_s$. Table A.1 shows the normalized kurtosis for some common source alphabets along with the limiting Gaussian values [8].

Real Valued Alphabet	Kurtosis	Complex Valued Alphabet	Kurtosis
Uniform BPSK	1	Uniform M-PSK	1
Uniform 4-PAM	1.64	Uniform 16-QAM	1.32
Uniform 8-PAM	1.762	Uniform 64-QAM	1.381
Uniform 16-PAM	1.791	Uniform 256-QAM	1.395
Uniform 32-PAM	1.798	Uniform 1024-QAM	1.399
Gaussian	3	Gaussian	2

Table A.1: Normalized source kurtosis for various source distributions

According to [8], non-uniform (i.e. shaped) source distributions have the effect of increased source kurtosis and as the source kurtosis approaches Gaussian, convergence to a desirable setting when using a stochastic gradient descent to find the CM minima (i.e. CMA) is practically impossible. For successful convergence, the source kurtosis needs to be *sub-Gaussian*, i.e. less than that of a Gaussian process.

Appendix B

Simulation Plots

B.1 Linear Equalization

B.1.1 Trained BS-LMS

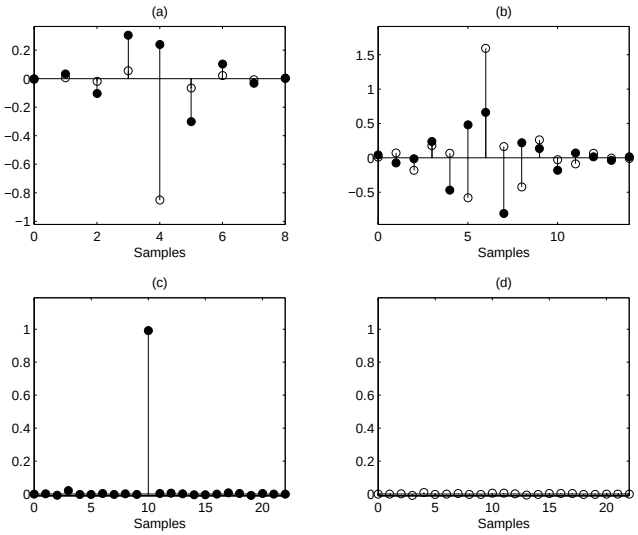

Figure B.1: Impulse responses for linear LMS equalization of Channel-A: (a) channel response, (b) equalizer response, (c) equalized response, (d) equalized response. real/imaginary $\Leftrightarrow$ filled/non-filled, SNR = 40 dB

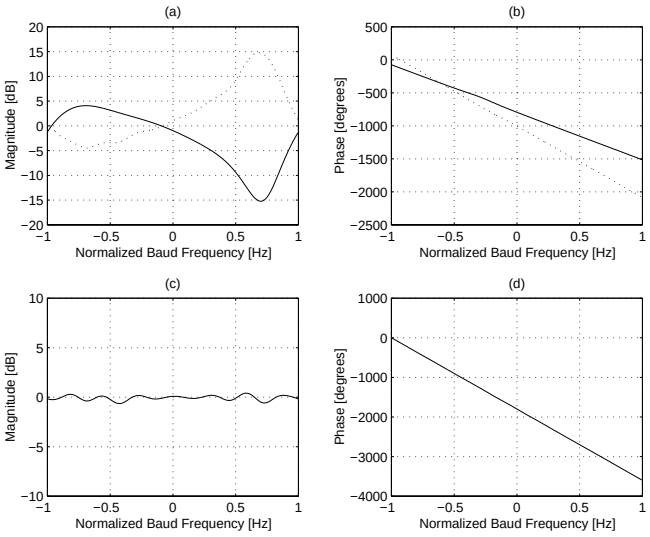

Figure B.2: Magnitude and phase responses for linear LMS equalization of Channel-A: (a) channel (solid) and equalizer (dotted) magnitude responses, (b) channel (solid) and equalizer (dotted) phase responses, (c) equalized magnitude response, (d) equalized phase response. SNR = 40 dB

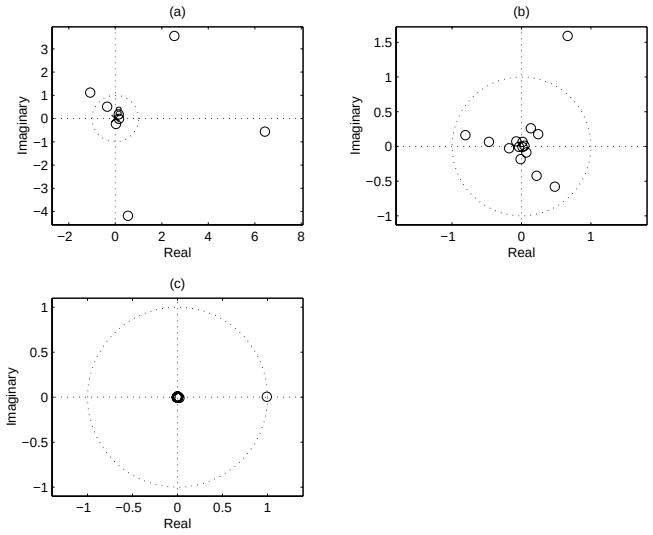

Figure B.3: Zeros for linear LMS equalization of Channel-A: (a) channel zeros, (b) equalizer zeros, (c) channel-equalizer combination zeros. SNR = 40 dB

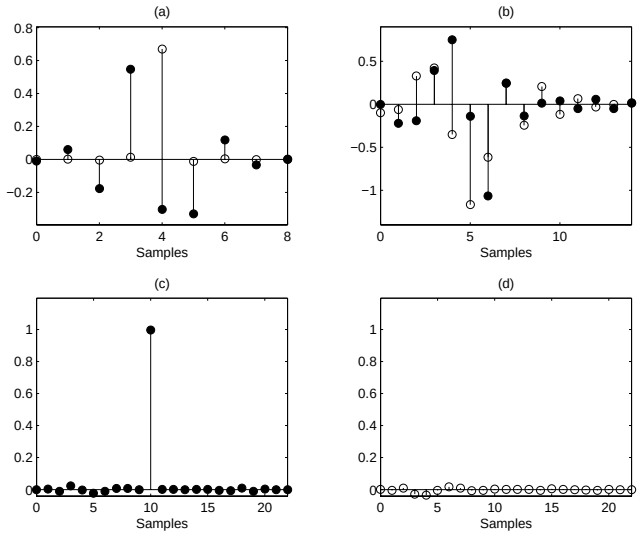

Figure B.4: Impulse responses for linear LMS equalization of Channel-B: (a) channel response, (b) equalizer response, (c) equalized response, (d) equalized response. real/imaginary $\Leftrightarrow$ filled/non-filled. SNR = 40 dB

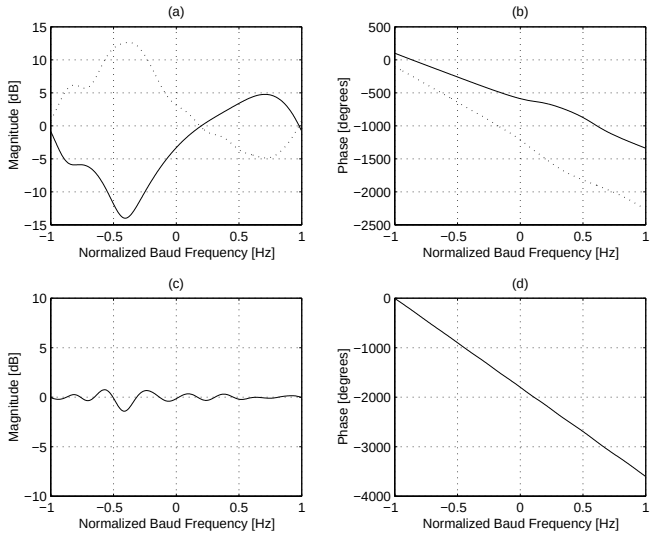

Figure B.5: Magnitude and phase responses for linear LMS equalization of Channel-B: (a) channel (solid) and equalizer (dotted) magnitude responses, (b) channel (solid) and equalizer (dotted) phase responses, (c) equalized magnitude response, (d) equalized phase response. SNR = 40 dB

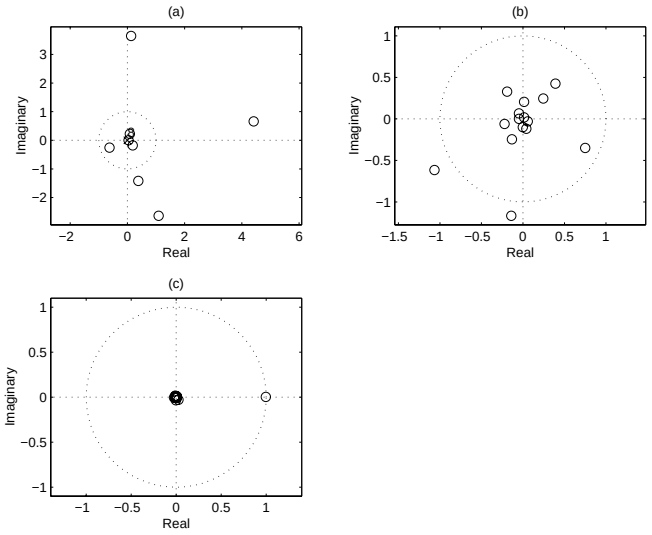

Figure B.6: Zeros for linear LMS equalization of Channel-B: (a) channel zeros, (b) equalizer zeros, (c) channel-equalizer combination zeros. SNR = 40 dB

B.1.2 Blind FS-CMA/DD

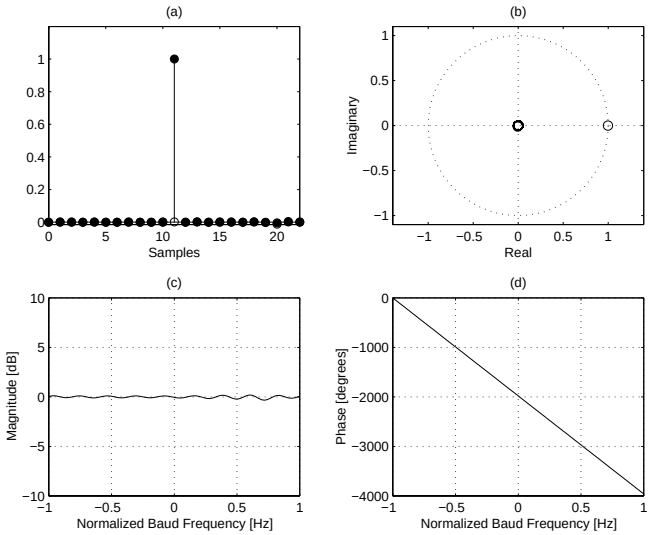

Figure B.7: Responses and zeros FS-CMA/DD linear equalization of Channel-A: (a) equalized BS impulse response (real/imaginary $\Leftrightarrow$ filled/non-filled), (b) equalized BS zeros, (c) equalized BS magnitude response, (d) equalized BS phase response. SNR = 40 dB

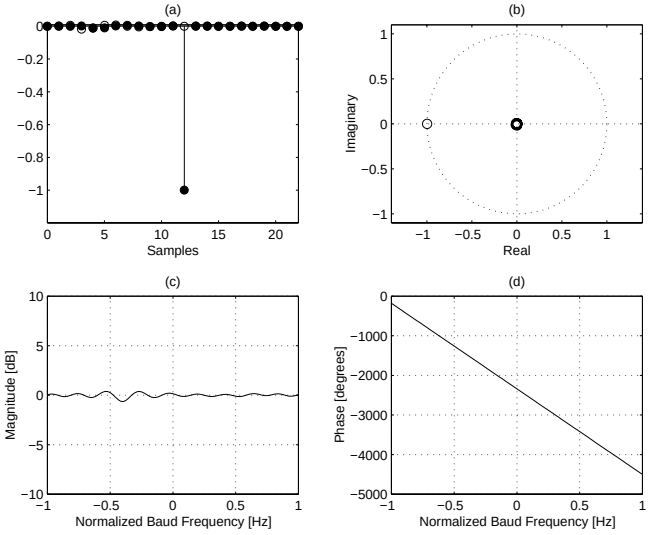

Figure B.8: Responses and zeros FS-CMA/DD linear equalization of Channel-B: (a) equalized BS impulse response (real/imaginary $\Leftrightarrow$ filled/non-filled), (b) equalized BS zeros, (c) equalized BS magnitude response, (d) equalized BS phase response. SNR = 40 dB

B.2 Decision Feedback Equalization

B.2.1 Trained BS-LMS-DFE

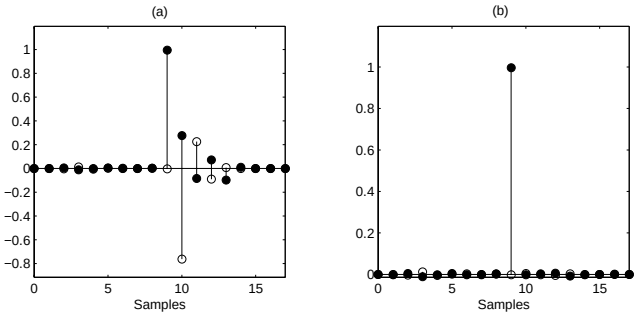

Figure B.9: Impulse responses for non-linear LMS-DFE and Channel-A: (a) channel-feedforward filter combination impulse response, (b) total impulse response including the feedback filter. (real/imaginary $\Leftrightarrow$ filled/non-filled), SNR = 40 dB

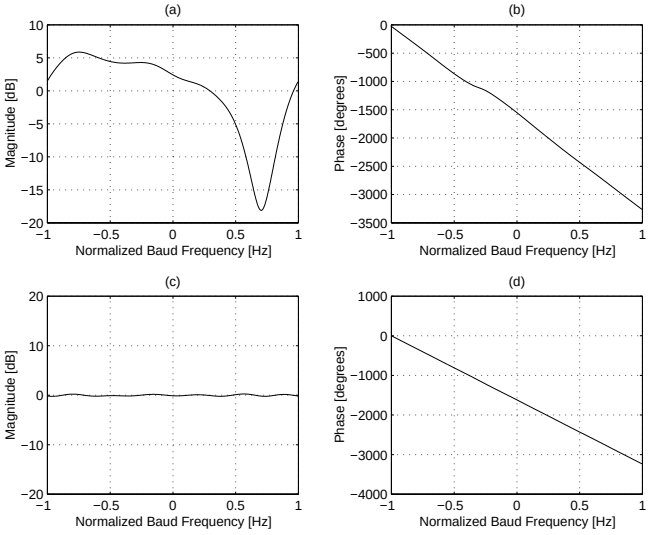

Figure B.10: Magnitude and phase responses for non-linear LMS-DFE and Channel-A: (a) channel-feedforward filter combination magnitude response, (b) channel-feedforward filter combination phase response, (c) total magnitude response including the feedback filter, (d) total phase response including the feedback filter. SNR = 40 dB

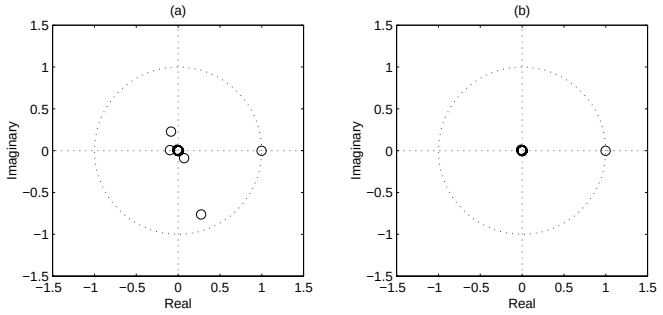

Figure B.11: Zeros for non-linear LMS-DFE and Channel-A: (a) channel-feedforward filter combination zeros, (b) equalized zeros including the feedback filter. SNR = 40 dB

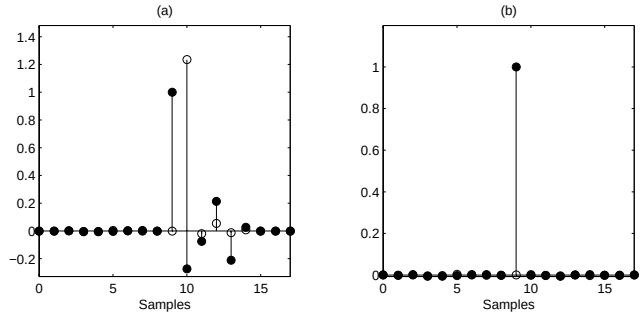

Figure B.12: Impulse responses for non-linear LMS-DFE and Channel-B: (a) channel-feedforward filter combination impulse response, (b) total impulse response including the feedback filter. (real/imaginary $\Leftrightarrow$ filled/non-filled), SNR = 40 dB

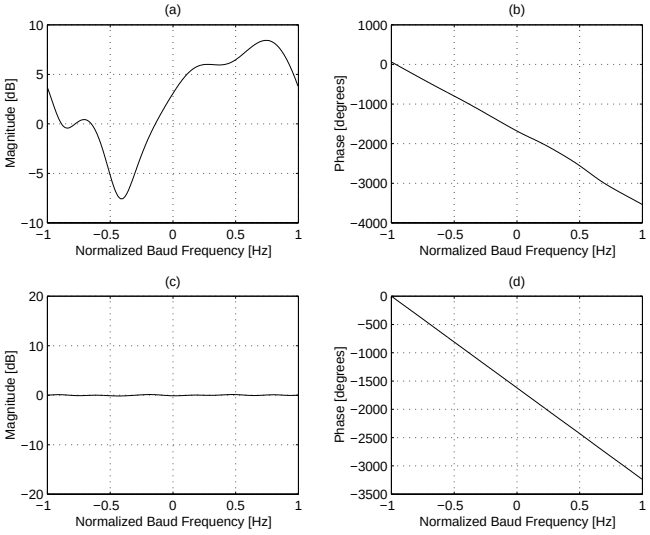

Figure B.13: Magnitude and phase responses for non-linear LMS-DFE and Channel-B: (a) channel-feedforward filter combination magnitude response, (b) channel-feedforward filter combination phase response, (c) total magnitude response including the feedback filter, (d) total phase response including the feedback filter. SNR = 40 dB

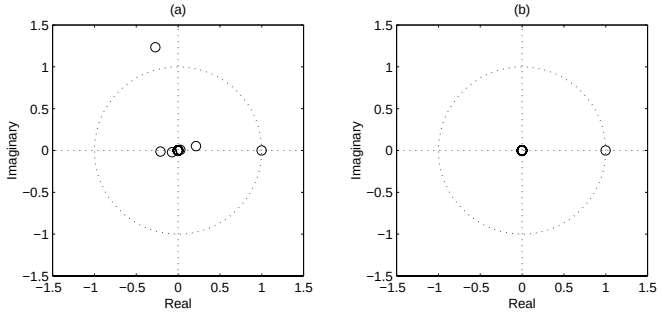

Figure B.14: Zeros for non-linear LMS-DFE and Channel-B: (a) channel-feedforward filter combination zeros, (b) equalized zeros including the feedback filter. SNR = 40 dB

B.2.2 Blind FS-CMA/DD-DFE

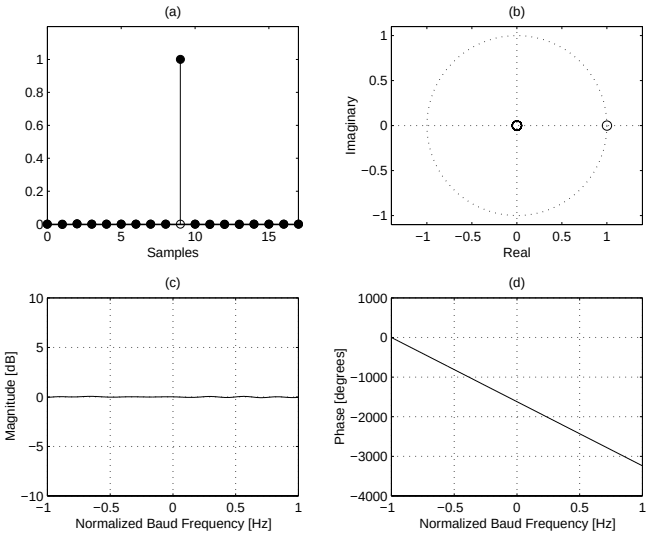

Figure B.15: Responses for FS-CMA/DD-DFE and Channel-A: (a) total channel-DFE filter combination impulse response (real/imaginary $\Leftrightarrow$ filled/non-filled), (b) total channel-DFE filter combination zeros, (c) total channel-DFE filter combination magnitude response, (d) total channel-DFE filter combination phase response. SNR = 40 dB

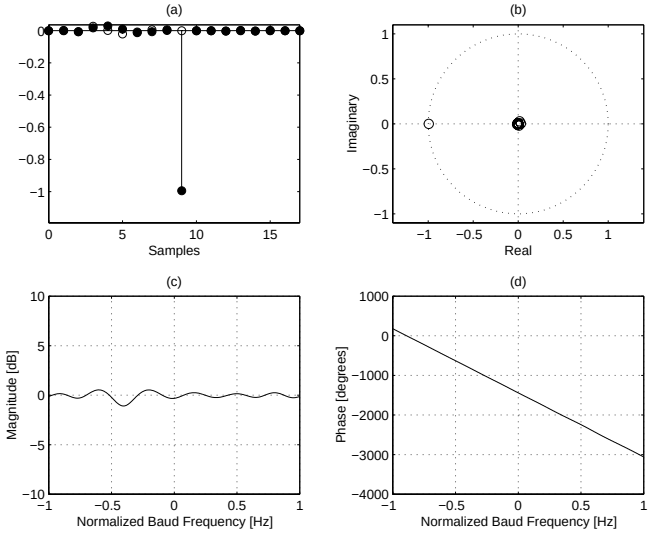

Figure B.16: Responses for FS-CMA/DD-DFE and Channel-B: (a) total channel-DFE filter combination impulse response (real/imaginary $\Leftrightarrow$ filled/non-filled), (b) total channel-DFE filter combination zeros, (c) total channel-DFE filter combination magnitude response, (d) total channel-DFE filter combination phase response. SNR = 40 dB

Bibliography

[1] L. Ahlin, J. Zander: *"Principles of Wireless Communications"*, second ed., Studentlitteratur, Lund, Sweden, 1998.

[2] J.W.M. Bergmans, A.J.E.M. Janssen: *"Robust Data Equalization, Fractional Tap Spacing and the Zak Transform"*, Philips Journal of Research, vol. 42, no. 4, pp. 351-398, 1987.

[3] R.A. Casas: *"Blind Adaptive Decision Feedback Equalization : a Class of Bad Channels"*, M.Sc. thesis, Cornell University, New York, USA, 1996.

[4] F.R.P. Cavalcanti, J.C.M. Mota: *"A Predictive Blind Equalizer Based in Godard's Criterion for 256-QAM Digital Radio Systems"*, Proceedings IEEE, 1996.

[5] R.D. Gitlin, J.F. Hayes, S.B. Weinstein: *"Data Communications Principles"*, Plenum Press, New York, USA, 1992.

[6] D.N. Godard: *"Self-recovering Equalization and Carrier Tracking in Two-Dimensional Data Communication Systems"*, IEEE Transactions on Communications, vol. 28, no. 11, pp. 1867-1875, Nov. 1980.

[7] C.R. Johnson et al.: *"The Core of FSE-CMA Behavior Theory"*, To appear as a chapter in the book *Unsupervised Adaptive Filtering*, Simon Haykin, Wiley, New York, USA, 1999.

[8] C.R. Johnson et al.: *"Blind Equalization Using the Constant Modulus Criterion: A Review"*, Proceedings IEEE, Oct. 1998.

[9] J.P. LeBlanc: *"Effects of Source Distributions and Correlation on Fractionally Spaced Blind Constant Modulus Algorithm Equalizers"*, Ph.D. dissertation, Cornell University, New York, USA, 1995.

[10] A. Leclert, P. Vandamme: *"Universal Carrier Recovery Loop for QASK and PSK Signal Sets"*, IEEE Transactions on Communications, pp. 130-136, Jan. 1983.

[11] E.A. Lee, D.G. Messerschmitt: *"Digital Communication"*, Kluwer Academic Publishers, Boston, USA, 1988.

[12] O. Macchi, E. Eweda: *"Convergence Analysis of Self-Adaptive Equalizers"*, IEEE Transactions on Information Theory, vol. IT-30, no. 2, 1984.

[13] T. Manning: *"Microwave Radio Transmission Design Guide"*, Artech House Books, USA, 1999.

[14] J.J. Olmos, R. Agusti, F.Casadevall: *"Decision Feedback Equalization and Carrier Recovery in 140 Mbit QAM Digital Radio Systems"*, Proceedings IEEE, 1998.

[15] J.G. Proakis: *"Digital Communications"*, third ed., McGRAW-HILL, USA, 1995.

[16] W.D. Rummler: *"A New Selective Fading Model: Application to Propagation Data"*, Bell Laboratories, 1978.

[17] J.R. Treichler, B.G. Agee: *"A New Approach to Multipath Correction of Constant Modulus Signals"*, IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-31, no. 2, pp. 459-472, Apr. 1983.

[18] J.R. Treichler, M.G. Larimore, J.C. Harp: *"Practical Blind Demodulators for High-order QAM signals"*, Proceedings of the IEEE special issue on Blind System Identification and Estimation, vol. 86, no. 10, pp. 1907-1926, Oct. 1998.

[19] T. Öberg: *"Modulation, Detektion och Kodning"*, Studentlitteratur, Lund, Sweden, 1998, (in Swedish).