Multimodulus Algorithms for Blind Equalization

by

Jian Yang

M.Sc. in Electrical Engineering, University of British Columbia, Vancouver, Canada

A Thesis Submitted in Partial Fulfillment of the

Requirements for the Degree of

PH.D IN APPLIED SCIENCE

in the

Department of Electrical Engineering

We accept this thesis as conforming

to the required standard

University of British Columbia

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Electrical Engineering

The University of British Columbia
Vancouver, Canada

Date Oct 23, 1997

DE-6 (2/88)

Abstract

In bandwidth-efficient digital transmission, the training of a receiver requires a start-up procedure. This start-up includes the three steps of setting the automatic gain control, recovering timing, and converging the adaptive filters. For many applications, start-up is facilitated by using a known training sequence, which can be used as an ideal reference by the receiver. However, sometimes the use of a training sequence is not feasible or not desirable. In this case, start-up has to be done blindly. The most challenging aspect of blind start-up is the convergence of the adaptive equalizer. This function is called blind equalization.

Without ideal reference, the receiver has to make decisions about what data have been transmitted. Normally we use a decision device to make the assumptions on the input signals. This decision device is also called a slicer. These decisions are highly unreliable because the received data are corrupted by intersymbol interference due to distortion introduced by the communication link. As a result, the equalizer does not converge with the conventional least mean square (LMS) algorithm because the probability of wrong decisions is too high. The requirement for reducing wrong decisions leads to the development of blind algorithms.

Several algorithms were developed in the past for blind equalization. The two most common ones are the constant modulus algorithm (CMA) and the reduced constellation algorithm (RCA). Both algorithms have been successfully used in many practical applications in data communications. However, improved blind algorithms are still in demand in order to meet the requirements of high data rate communications, where the implementation of blind equalization is required to be simple and reliable.

RCA is simple to implement but is not very reliable. CMA is very reliable but has a high complexity. In this thesis, an improved blind algorithm, called multimodulus algorithm (MMA), is proposed. This new algorithm is more reliable than RCA and less complex than CMA. Blind algorithms, such as RCA or CMA, do not take full advantage of the knowledge of the statistics of the data signal, because they only use one statistical quantity, called modulus, during blind convergence. In contrast, MMA makes better usage of the knowledge of the statistics of the data signal and employs multiple moduli to achieve initial convergence. The use of multiple moduli

makes MMA very flexible and easily adaptable to applications using two-dimensional transmission schemes with nonsquare or very dense signal constellations. Both RCA and CMA are not very effective for these types of applications.

Most blind equalization algorithms minimize a cost function. Our investigation has shown that the minimum (residual) value of this cost function has a great influence on the reliability and speed of convergence of the corresponding blind algorithm. For example, for CMA, RCA and MMA, this residual value increases when the bandwidth efficiency of the transmission system increases, and the blind algorithms suffer a corresponding decrease in performance.

Two generalized versions of MMA are proposed to circumvent the above problem. One is called generalized MMA (GMMA). The basic version of MMA uses multiple moduli primarily for non-square constellations when nonuniform symbol distributions are utilized. Multiple moduli are also used with GMMA but the additional purpose there is to decrease the residual value of the cost function.

A similar effect is achieved by using another blind algorithm, called windowed MMA (WMMA). With WMMA, only one modulus is used, but the tap coefficients of the filter are only updated with some of the data, which leads to a reduction of the residual value of the cost function.

The new MMA algorithm has been extensively tested with computer simulations. Most simulation results presented here were obtained with the 155 Mb/s 64-CAP transceiver specified in the ATM LAN standard for category 3 unshielded-twisted-pair office wiring. However, our results are equally applicable to other applications, such as the 52 Mb/s 16-CAP transceiver defined in DAVIC's specification for fiber-to-the-curb networks. Some preliminary experimental results obtained in the laboratory with such a 16-CAP transceiver are presented in this thesis. The computer simulations and laboratory experiments have confirmed that the basic MMA and its generalized algorithms show improved blind convergence performance or complexity savings when compared to previously available algorithms. MMA will likely be incorporated in future broadband access systems from Lucent Technologies, and will replace RCA and CMA, which were previously used, or eliminate the need for a training sequence.

Contents

List of Tables

List of Figures

List of Abbreviations

ADSL: asymmetric digital subscriber line

A/D: analog-to-digital (converter)

AGC: automatic gain control

AWG: American wire gauge

ATM: asynchronous transfer mode

BALUN: balance/unbalance

CAP: carrierless amplitude modulation/phase modulation

CMA: constant modulus algorithm

DAB: digital audio broadcast

DAVIC: digitial audio-visual council

D/A: digital-to-analog (converter)

DFE: decision feedback equalizer

DSP: digital signal processing (or processor)

HDTV: high definition TV

FIFO: first-in-first-out

FIR: finite impulse response

FSLE: fractionally spaced linear equalizer

FTTC: fiber-to-the-curb

GMMA: generalized multimodulus algorithm

ISI: intersymbol interference

LAN: local area network

LMS: least mean square

LPF: low pass filter

MMA: multimodulus algorithm

MSE: mean squared error

NEXT: near end crosstalk

NRZ: non-return to zero

QAM: quadrature amplitude modulation

RCA: reduced constellation algorithm

RF: radio frequency

SDV: switched digital video

2B1Q: two-binary one-quaternary

UTP: unshielded twisted pair

WMMA: windowed multimodulus algorithm

Acknowledgements

I would like to thank my supervisor Professor Guy A. Dumont for his help and support during the course of this work.

I would like to thank my Technical Manager Dr. J.J. Werner who benefited me greatly from his twenty years of experience in data communications at Bell Labs, and my Director Dr. Victor Lawrence for his support of this work.

I would like to thank my colleague Dale Harman who contributed to the experimental lab work with the real-time and real world DSP setups. Thanks also to the other members of the Advanced Data Communications Group of Bell Labs Innovations for their help and encouragements.

Chapter 1

Introduction

In many applications of data communications, blind algorithms are required to provide initial convergence of the adaptive equalizer. Blind algorithms behave differently from traditional adaptive algorithms, because they provide filter adaptation without accurate reference signals. In the next section, we describe some applications where blind equalization is required or desirable. Section 1.2 discusses some fundamental concepts, such as bandwidth efficiency, CAP line codes, and blind equalization. Finally, a literature review and a thesis review are provided at the end of the chapter.

1.1 Applications of Blind Equalization

Asynchronous transfer mode (ATM) local area network (LAN) standards for telephone wiring in office buildings have recently been finalized [1]. These standards are different from previous LAN standards in that they use bandwidth-efficient digital transmission schemes. For such transmission schemes, the initial training of a receiver consists of the three main steps of setting the automatic gain control (AGC), recovering timing (clock synchronization), and converging the adaptive filters. This initial training is called start-up. For many applications, start-up is facilitated by using a training sequence. With a training sequence, an equalizer receives data that consist of known patterns also called ideal references. When the transmitted data are known, the equalizer can calculate meaningful errors with respect to the equalizer's outputs because filter adaptation is done with an ideal reference. However, when the use of a training sequence is not feasible or not

desirable, the adaptive filters require a blind start-up. Without ideal reference, filter adaptation is called blind equalization. The major disadvantage of blind equalization is that it takes much more time to converge an equalizer than is feasible with an ideal reference.

The decision to use blind equalization depends on the applications. Sometimes the use of blind equalization is just an option and may be chosen for cost reasons. However, due to environment realities and systems considerations in networks, the use of blind equalization may often be the only choice. The following gives examples of applications which typically use blind equalization.

- *Point-to-Point Networks*:

Fig. 1.1(a) shows a point-to-point communication link used for ATM-LAN applications [17] [18]. Communication between the ATM LAN switch and the local workstations is done on two separate twisted pairs. The usage of a training sequence is not specified in any of the ATM LAN standards which have been recently developed. Thus, when the workstation in Fig. 1.1(a) is powered on, both receivers must go through a blind start-up.

- *Broadcast Networks*

In the broadcast network shown in Fig. 1.1(b), the same signal is received by a multiplicity of receivers. This type of network topology will be used in forthcoming cable TV and wireless applications, such as high definition TV (HDTV) and digital audio broadcast (DAB). The number of receivers that will be involved in these applications can range from several hundreds to several millions. Under these conditions, it is not feasible for the transmitter to send a dedicated training sequence to each receiver. HDTV and DAB solve the start-up problem by periodically sending a training sequence, which results in a reduction of the actual data rate available to the user. For this and other reasons, it is expected that a periodic training sequence will not be made available for forthcoming cable TV applications, so that the receivers will need a blind start-up.

- *Point-to-Multipoint Networks*

The configuration in Fig. 1.1(c) is used for voiceband modems. More recently, this network topology has also been adopted for local distribution in fiber-to-the-curb (FTTC) network architectures used for switched digital video (SDV) systems [1] [5] [13]. In Fig. 1.1(c), the transmitter on the left continuously transmits the same downstream signal to all the receivers on the right. The upstream

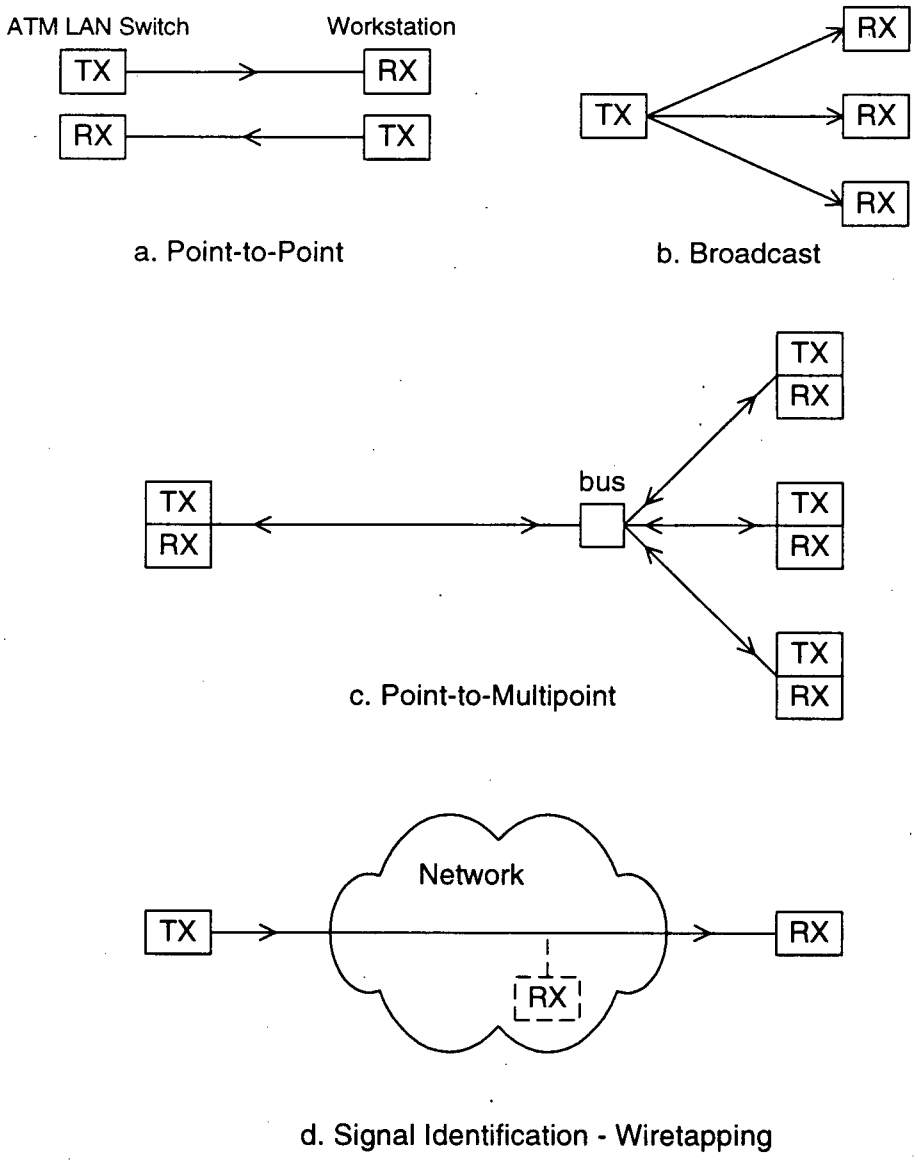


Figure 1.1: Applications of blind equalization

channel, in the opposite direction, is shared by all the transmitters on the right. Transmission in this direction is done in a bursty fashion. That is, each transmitter on the right transmits for a short period of time, when requested to do so by the downstream channel, and then stops. The upstream burst modem has to start up very fast in order to minimize the overhead associated with start-up. Thus, a training sequence is required. On the other hand, blind start-up is mandatory for the downstream channel if one wants to avoid the transmission of a periodic training sequence. To see why, we can consider the FTTC application, for example. In this case, the transceivers on the right in Fig. 1.1(c) are located in set-top boxes (or PCs). When one of the set-top boxes is suddenly turned on the corresponding receiver has to be trained. Sending a training sequence to this receiver in the downstream channel would disrupt transmission to the other set-top boxes, which is unacceptable. Thus, the receiver has to start blindly.

- *Signal Identification-Wiretapping*

The last application of blind equalization discussed here is signal identification [3]. For a variety of reasons, it may be desirable to tap a communication link and try to identify the kind of signal that is being transmitted. This is being done, for example, for signal classification purposes in the telephone network, as is graphically illustrated in Fig. 1.1(d). A general purpose receiver is fed the tapped signal and first determines whether the signal on the telephone connection is a speech signal or a data signal. If it is a data signal, it tries to blindly equalize this signal and determine its type from the equalized output signal and some other parameters.

In summary, blind equalization is desirable in many applications because it reduces complexity and may be the only alternative available. Popular blind equalization algorithms have been developed for many years. However, improved algorithms for blind equalization are still in high demand in high-rate data communications.

1.2 Bandwidth-Efficient Transmission Schemes

Advanced techniques are required in order to meet the demand for high data rate communications. In ATM LAN applications using category 3 wiring, for example, bandwidth-efficient transmission schemes are now being used to increase the data rate available on the desktop [1]. In the LAN

community, transmission or modulation schemes are also called line codes, and we will occasionally use this expression. Bandwidth efficiency for high data rates ($\geq$50 Mb/s) over category 3 cables in office buildings is required because the useful bandwidth of a 100m category 3 cable is about 30 MHz, and because there are stringent radiation limit requirements above 30 MHz. The bandwidth efficiency I is defined as

$$I = \frac{R}{W} \tag{1.1}$$

where R is the data rate, expressed in bits per second (bps), W is the bandwidth utilization of the data signal, expressed in hertz (Hz), and the bandwidth efficiency I is expressed in bps/Hz. We see that the bandwidth efficiency can be improved by either increasing R for a given W, or by decreasing W for a given R. This can be done by using two independent techniques, which are multilevel encoding and efficient (Nyquist) spectral shaping.

1.2.1 Multilevel Encoding

The first technique that can be used to increase bandwidth efficiency is multilevel encoding. In the following, we show how the bandwidth efficiency is increased by using this technique. Examples are given for one-dimensional (1-D) and two-dimensional (2-D) transmission schemes. Special attention is given to a 2-D scheme called carrierless amplitude modulation/phase modulation (CAP).

- *One-dimensional multilevel mapping*

The simplest transmission scheme available uses binary encoding, in which two possible symbol levels (± 1) are used to transmit one bit of information. With multilevel encoding, blocks of m bits are mapped into symbols that have more than two values. Fig. 1.2 shows examples of binary encoding and multilevel encoding. With the non-return-to-zero (NRZ) line code shown in Fig. 1.2 (a), each single bit is encoded into ± 1 values. With the two-binary one-quaternary (2B1Q) line code shown in Fig. 1.2 (b), blocks of two bits are encoded into four possible values proportional to $\{\pm 1, \pm 3\}$. In general, the encoding of blocks of m bits requires the usage of 2^m different symbol values. For a given bit rate R, multilevel encoding reduces the bandwidth requirements by a factor of m because the width of the square pulses used by the line codes is increased by m. Fig. 1.3 shows the Fourier transforms of the pulses used for the NRZ and 2B1Q transmission schemes, when they

provide the same data rate R.

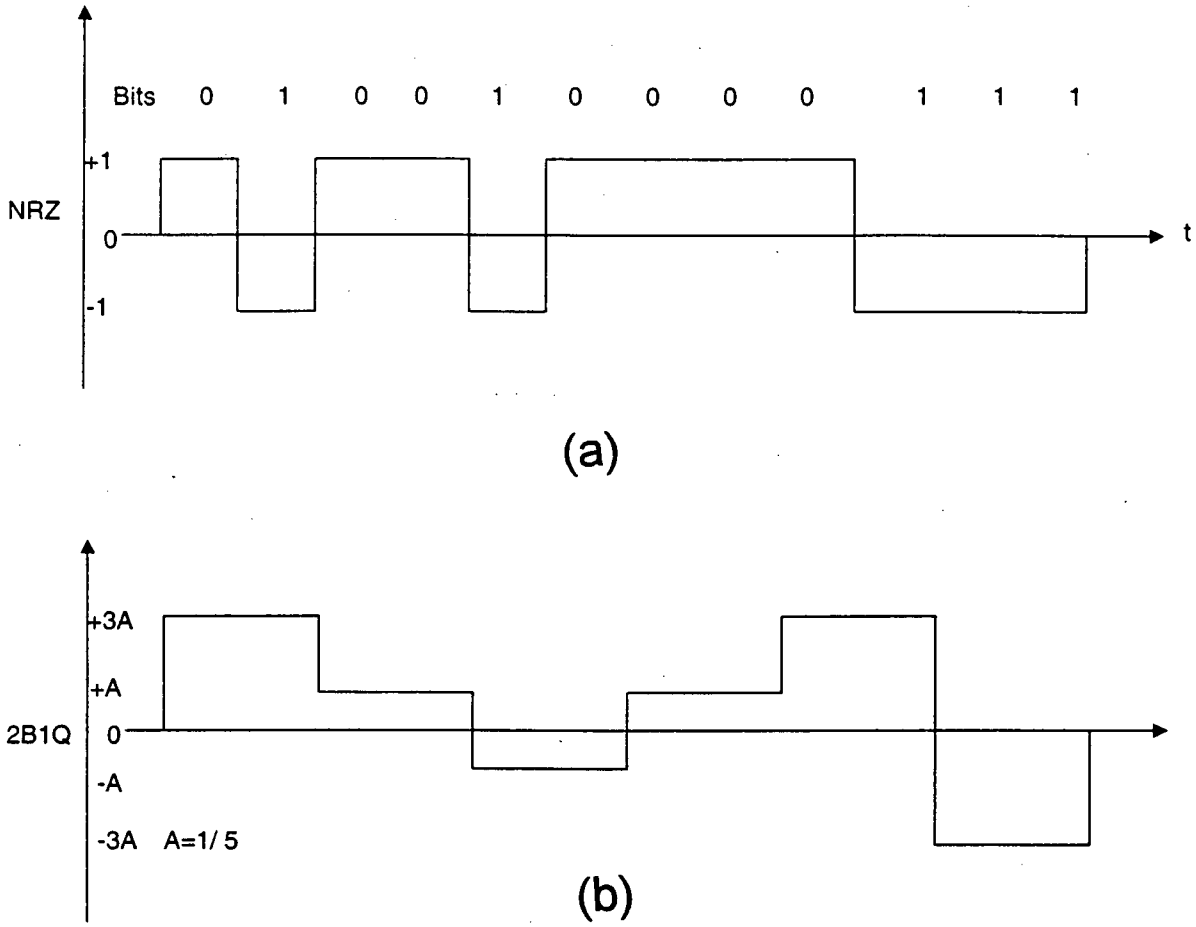
It is usually assumed that the useful bandwidth W of NRZ and 2B1Q extends up to the first null. For NRZ this first null occurs at a frequency which is equal to the bit rate R, so that $R = W$ and the bandwidth efficiency of NRZ is 1 bps/Hz. For 2B1Q we have $W = \frac{R}{2}$, so that the bandwidth efficiency is 2 bps/Hz. The rate at which the square pulses in Fig. 1.2 are sent through the channel is called symbol rate. We see that half the symbol rate is needed for 2B1Q compared to NRZ. Thus, in general, if $1/T$ and W are the symbol rate and bandwidth required for binary encoding, then the usage of multilevel encoding reduces these requirements to $1/mT$ and W/m. Therefore, one-dimensional digital transmission schemes can potentially provide a bandwidth efficiency close to m bps/Hz by using multilevel encoding only. It will be shown in section 1.2.2 that this bandwidth efficiency can be further increased to $2m$ bps/Hz by using efficient spectral shaping.

- *Two-dimensional encoding and signal constellation*

Two-dimensional mapping is another technique which can be used to increase bandwidth efficiency. As shown in Fig. 1.5 (a), blocks of bits are first mapped into two separate pulse streams. After convolving with two shaping filters, the two pulse streams are then added for transmission over the channel. In this case, the two shaping filters are required to have some well-defined properties. For two-dimensional encoding, they are required to have the same spectral amplitude characteristics, and impulse responses which are orthogonal to each other.

When two-dimensional encoding is used, it is convenient to introduce the concept of signal constellation, which is the display of all the possible discrete symbols in the complex plane [27]. The signal constellations used for some LAN applications are shown in Fig. 1.6. For instance, for 64-CAP, blocks of six bits (i.e. $m = 4$) are represented by sixty four possible complex symbols or two-dimensional points. One way of doing this is to map the first three bits into symbols a_n and the last three bits into symbols b_n. Three bits can represent eight decimal numbers or symbol levels. In the ATM LAN standard, we use odd integer numbers to represent the symbol levels, and the bit-to-symbol mapping is as follows

$$000 \rightarrow 1 \quad 001 \rightarrow 3 \quad 010 \rightarrow 5 \quad 011 \rightarrow 7 \quad 100 \rightarrow -1 \quad 101 \rightarrow -3 \quad 110 \rightarrow -5 \quad 111 \rightarrow -7 \tag{1.2}$$



- Multilevel encoding maps blocks of bits into multilevel pulses
- The encoding of blocks of m bits requires 2^m different pulse levels

Figure 1.2: Multiple level encoding

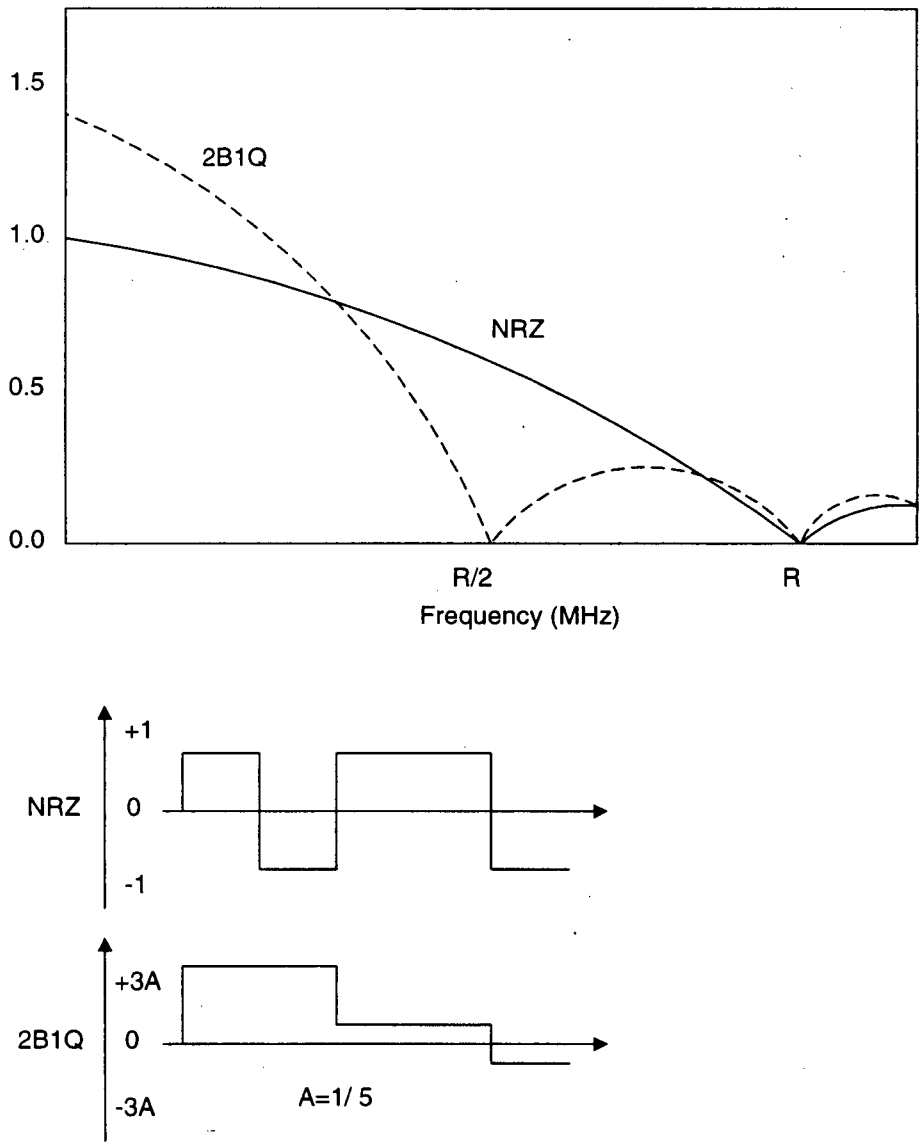


Figure 1.3: NRZ and 2B1Q line code

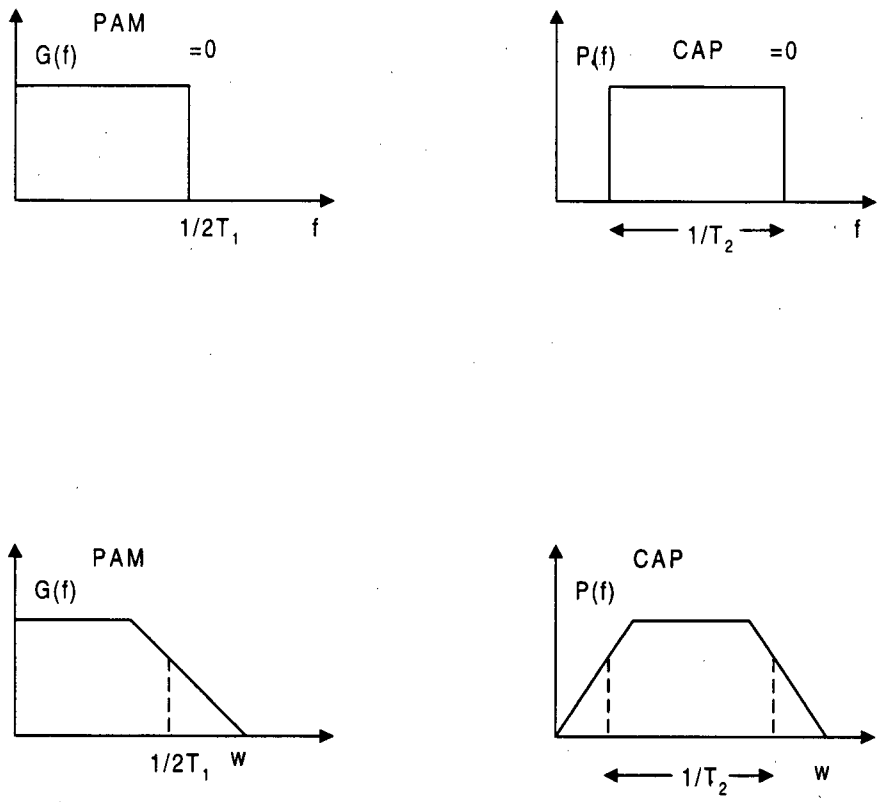


Figure 1.4: Definition of excess bandwidth

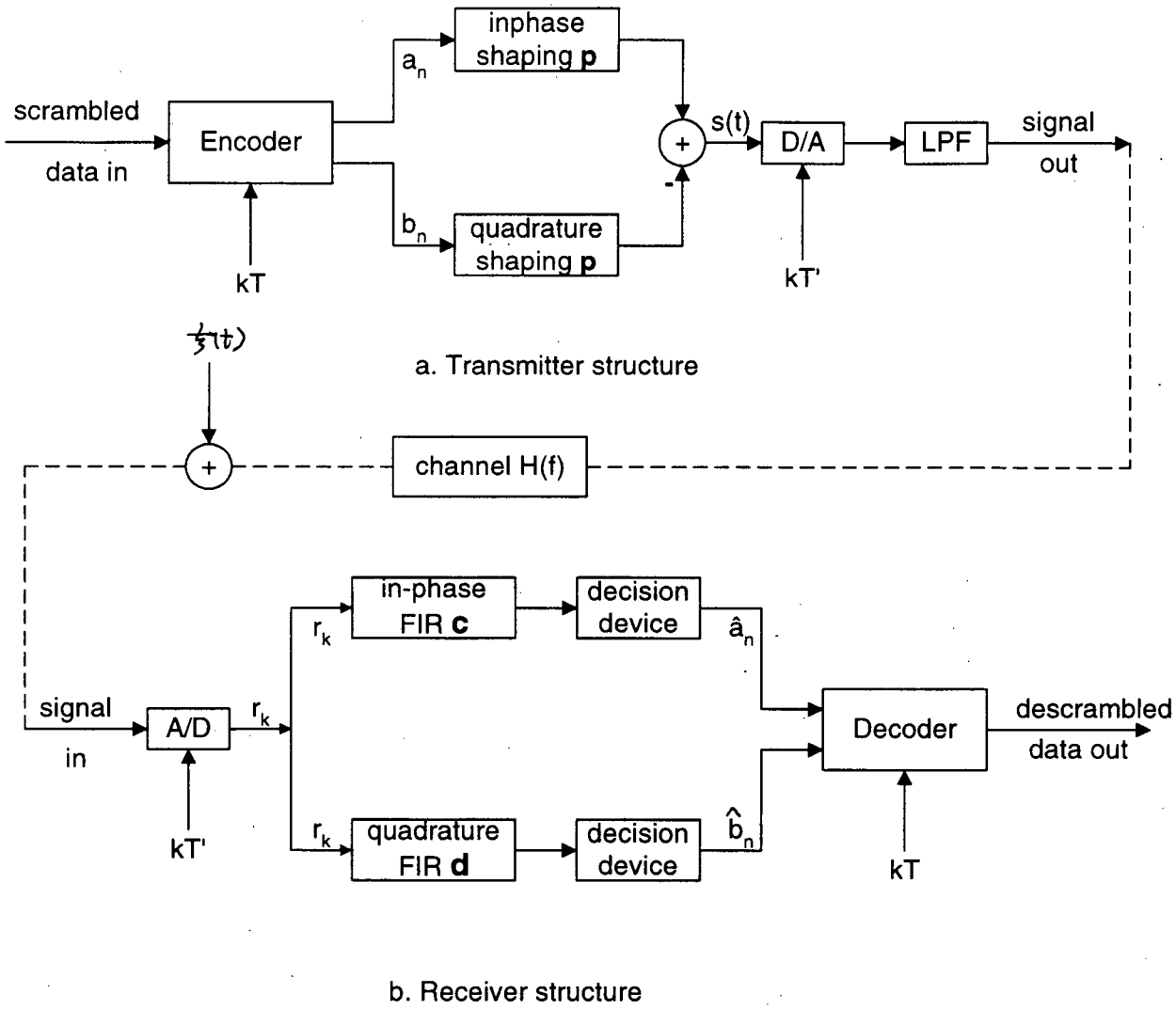


Figure 1.5: Communication link using CAP

An example of a bit sequence and of the corresponding transmitted symbols is given below

$$\text{bit sequence} \quad : \quad 0\,1\,1\,0\,1\,0\,1\,0\,1\,0\,0\,0 \cdots \tag{1.3}$$

$$\text{symbol } a_n \quad : \quad 7 \quad | \quad -3 \quad | \cdots \tag{1.4}$$

$$\text{symbol } b_n \quad : \quad 5 \quad | \quad 1 \quad | \cdots \tag{1.5}$$

The two-dimensional symbols $\{a_n, b_n\}$ can be represented as two-dimensional (or complex) points in the (a_n, b_n) plane. We call them real and imaginary symbols, or in-phase and quadrature phase symbols, and the symbols $A_n = a_n + jb_n$ are called complex or 2-D symbols. Since each symbol a_n and b_n takes one of eight possible values $\{\pm 1, \pm 3, \pm 5, \pm 7\}$, the display of all the possible combinations $\{a_n, b_n\}$ requires a total of $8 \times 8 = 64$ points, as shown in Fig. 1.6. With 2-D multilevel encoding only, it is possible to achieve a bandwidth efficiency of $\frac{m}{2}$ bps/Hz. This can be increased to m bps/Hz by using efficient spectral shaping, which will be discussed in Section 1.2.2.

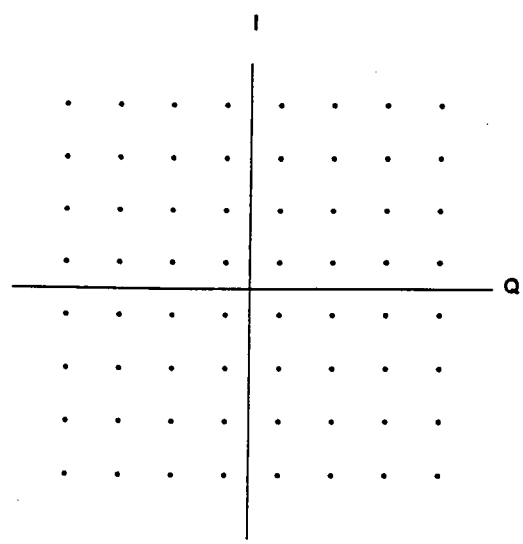


Figure 1.6: Signal constellations for two-dimensional encoding

- *Encoding improvement factors*

We now provide formal definitions for the improvement factors in bandwidth efficiency that result from multilevel encoding [27]. Multilevel encoding reduces the bandwidth requirement because it allows a decrease in symbol rate for a given bit rate R. Specifically, the reduction in symbol rate is equal to the number of bits transmitted in each symbol period. Pulse amplitude modulation (PAM)

is a one-dimensional transmission scheme. If m_1 bits are mapped into one-dimensional symbols for PAM then the symbol rate is reduced by a factor m_1 with respect to 2-PAM, which is a 2-level PAM line code (NRZ is an example of a 2-PAM system with $m_1 = 1$). If m_2 bits are mapped into two-dimensional symbols for CAP then the symbol rate is reduced by a factor $m_2/2$ with respect to 4-CAP, which is a 4-point CAP line code ($m_2 = 2$). Since m_1 and m_2 are also the ratios between the bit rates R and the symbol rate $1/T_1$ and $1/T_2$, respectively, the improvement factors IF_{enc} in bandwidth efficiency due to multilevel encoding alone can be written as

$$IF_{enc} = \frac{R}{1/T_1} = m_1 \quad \text{for } 1-\text{D line codes} \tag{1.6}$$

$$IF_{enc} = \frac{R}{1/T_2} = \frac{m_2}{2} \quad \text{for } 2-\text{D line codes} \tag{1.7}$$

1.2.2 Efficient Spectral Shaping

The second technique that can be used to increase bandwidth efficiency is efficient spectral shaping. With this technique, the square pulses used for NRZ and 2B1Q in Fig. 1.7, which have one symbol period duration, are replaced with smoother pulses whose duration is several symbol periods. Examples of such pulses will be provided in later sections. With efficient spectral shaping, the minimum bandwidth W_{min} that can be used for the transmission of the signals is given by [27]:

- $W_{min} = \frac{1}{2T_1}$ for 1-D transmission schemes using a symbol rate $\frac{1}{T_1}$
- $W_{min} = \frac{1}{T_2}$ for 2-D transmission schemes using a symbol rate $\frac{1}{T_2}$

The amount of bandwidth used in excess of the theoretical minimum is called excess bandwidth (EBW), as shown in Fig1.4. Let W be the actual bandwidth utilization, we then have

$$EBW_1 = \frac{W - W_{min}}{W_{min}} = \frac{W - 1/2T_1}{1/2T_1} \quad \text{for } 1-\text{D schemes} \tag{1.8}$$

$$EBW_2 = \frac{W - W_{min}}{W_{min}} = \frac{W - 1/T_2}{1/T_2} \quad \text{for } 2-\text{D schemes} \tag{1.9}$$

where the excess bandwidth is expressed as a percentage of the minimum bandwidth. Thus, a system that uses the theoretical minimum bandwidth has zero excess bandwidth, and a system that uses twice the theoretical minimum bandwidth has 100% excess bandwidth (EBW=1).

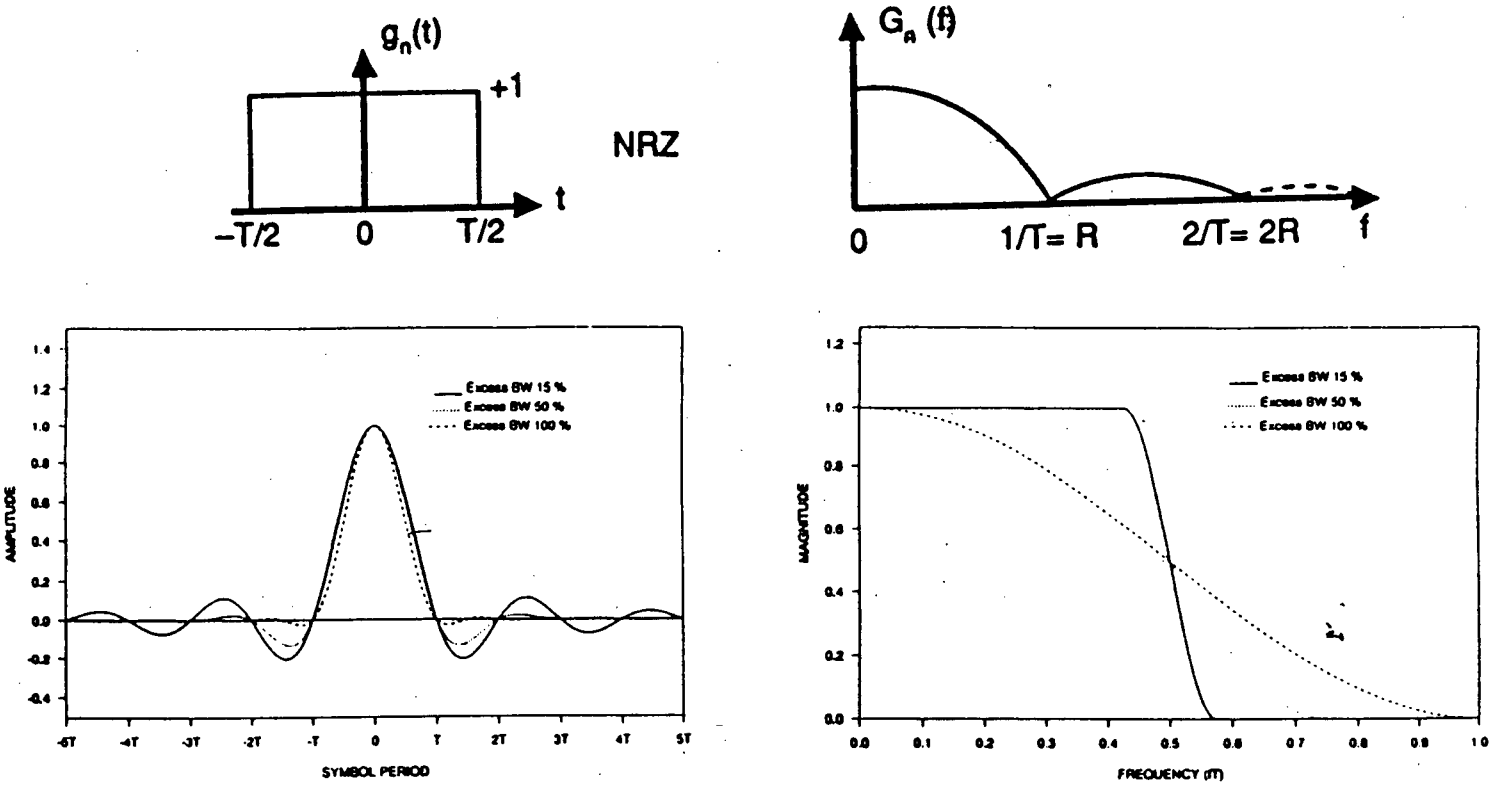


Figure 1.7: Efficient spectral shaping

1.2.3 Maximum Bandwidth Efficiency

Multilevel encoding and efficient spectral shaping are two techniques which can be used independently to improve bandwidth efficiency. The best results are obtained when the two techniques are combined. In this case, the maximum bandwidth efficiency that can be achieved for 1-D and 2-D line codes is given by

$$I_{1,max} = \frac{R}{W_{min}} = \frac{R}{1/2T_1} = 2m_1 \tag{1.10}$$

$$I_{2,max} = \frac{R}{W_{min}} = \frac{R}{1/T_2} = m_2 \tag{1.11}$$

Given a 1-D line code, it is always possible to design a 2-D line code which provides the same bit rate R and uses the same bandwidth W. This can be done by mapping twice the number of bits in each complex symbol, i.e., $m_2 = 2m_1$, and using a symbol rate that is half that of the 1-D line code, i.e., $1/T_2 = 1/2T_1$. However, it is not always straightforward to design a 1-D line code which is equivalent to a given 2-D line code. For example, for 32-CAP, we have $m_2 = 5$, and an

equivalent 1-D line code would require $m_1 = \frac{m_2}{2} = 2.5$, which is not easily implementable.

1.3 Digital CAP Transceivers

This section discuss the digital CAP transceivers, which are used in ATM LAN and other applications.

- *CAP versus QAM*

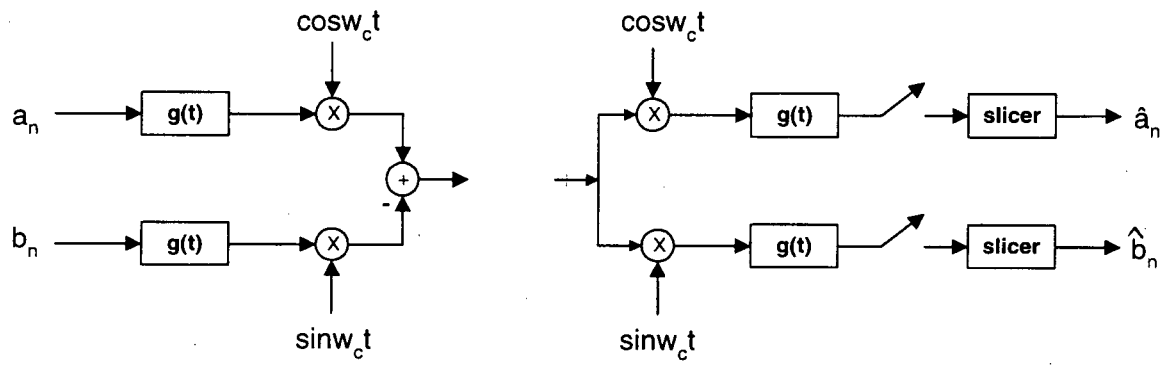
Carrierless amplitude modulation/phase modulation (CAP) is a bandwidth-efficient two-dimensional passband line code. CAP is closely related to the more familiar quadrature amplitude modulation (QAM) transmission scheme. In voiceband modems, QAM has been used for over 25 years, while CAP has been used for over 15 years. However, CAP is simpler to implement digitally. Transceiver structures for the CAP and QAM transmission schemes are shown in Fig. 1.8. Both schemes use two-dimensional encoding. With the conventional QAM transceiver structure shown in Fig. 1.8 (a), a modulator and a demodulator are required at the transmitter and receiver. The two modifications for QAM that lead to a CAP system are shown in Figs. 1.8 (b) and 1.8(c) [4]. The first modification is to replace the low-pass filters in Fig. 1.8 (a) with in-phase and quadrature passband shaping filters, as shown in Fig. 1.8 (b). In order to remain compatible with the QAM system in Fig. 1.8 (a), it is necessary to rotate the symbols at the symbol rate before feeding them to the shaping filters. A similar rotator must also be used at the receiver. Note that the rotator will be discussed in Section 3.1.2 and here we simply express the equation of how to computing the rotator

$$\theta = Im\{Y_n * A_n*\} \tag{1.12}$$

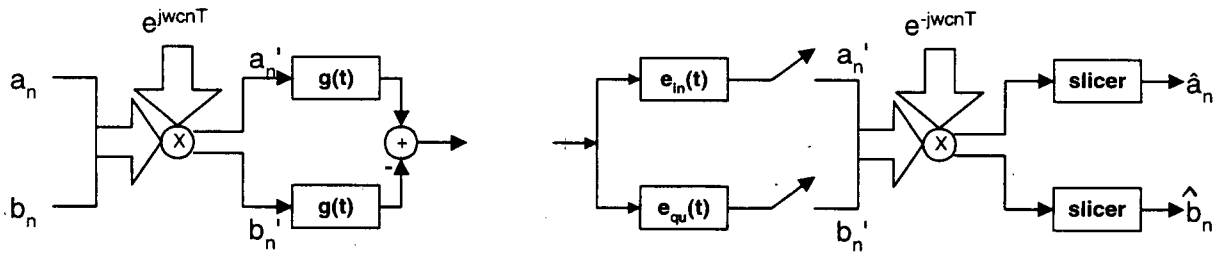
The next modification, which leads to CAP, is to remove the rotators at the transmitter and receiver, as shown in Fig. 1.8 (c). This CAP system will generally not be compatible with the structures shown in Fig. 1.8 (b) and Fig. 1.8 (c), but provides the same theoretical performance, while being simpler to implement.

- *Structure of CAP transceivers*

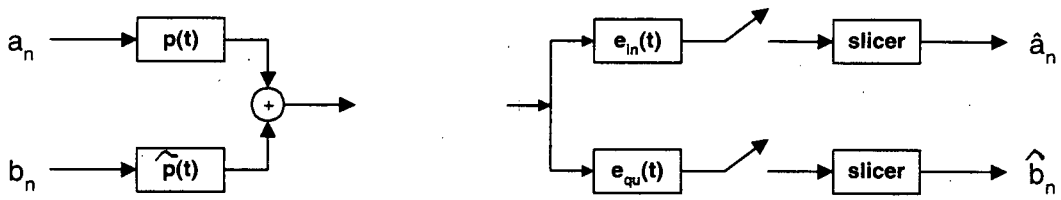
The structure defined by the ATM LAN standard for a digital CAP transceiver is illustrated in Fig1.5. At the input of the transmitter, the scrambled data are fed into an encoder, where



a. Conventional QAM transceiver structure



b. Modified QAM transceiver structure



c. CAP transceiver structure

Figure 1.8: CAP versus QAM

the bit-to-symbol mapping is performed by a two-dimensional multilevel encoder. The encoder maps blocks of bits into two-dimensional multilevel symbol streams a_n and b_n. For instance, for the 64-CAP application, three bits are mapped into eight-level symbols in each dimension, that is, $\{a_n\} = \{b_n\} = \{\pm 1, \pm 3, \pm 5, \pm 7\}$. Note that odd integers are used as symbol levels for the ATM standard [1]. A 64-CAP signal constellation is shown in Fig. 1.6 (b).

The encoded symbols are then fed into two parallel shaping filters, one is an in-phase filter p_n and the other one is a quadrature phase filter $\tilde{p}_n$. Details for the design of the shaping filters can be found in Appendix A or [18] and [27]. Typically, the shaping filters have square-root raised cosine characteristics. The impulse responses of the shaping filters for the ATM LAN 64-CAP standard are plotted in Fig. 1.9. Figs. 1.10 (a) and (b) show the frequency responses of the shaping filters for baseband and passband, respectively. The 64-CAP transceiver uses a symbol rate $1/T$=25.92 Mbaud, and a center frequency f_c=15 MHz. Since each 64-CAP symbol represents 6 bits, we have a bit rate $R = 25.92 \times 6 = 155.52$ Mb/s. The bandwidth efficiency achieved by the 64-CAP system is equal to: $I = 155.52/30 \approx 5.2$ bps/Hz.

The two shaping filters of a CAP transceiver have impulse responses which form a Hilbert pair. The transfer function of a filter that produces a Hilbert transform is given by $-jsgn(f)$. That is, the two shaping filters have the following relation

$$P_Q(f) = -jsgn(f)P_I(f) \tag{1.13}$$

where $P_I(f)$ and $P_Q(f)$ are the transfer functions of the in-phase and quadrature phase shaping filters respectively. The outputs of the two shaping filters in Fig. 1.5 are subtracted (or added) and the result is then fed into a digital-to-analog (D/A) converter followed by an interpolating low-pass-filter (LPF). The analog CAP signal that is launched on the line can be expressed as

$$s(t) = \sum_{n=-\infty}^{\infty} a_n p(t - nT) - \sum_{n=-\infty}^{\infty} b_n \tilde{p}(t - nT) \tag{1.14}$$

where $p(t)$ and $\tilde{p}(t)$ are the overall impulse responses of the cascades of shaping filters, D/A, and low-pass filter. In the case of the ATM standard, the signal $s(t)$ is transmitted through unshielded twisted pair (UTP) category 3 cables with lengths up to 100 meters.

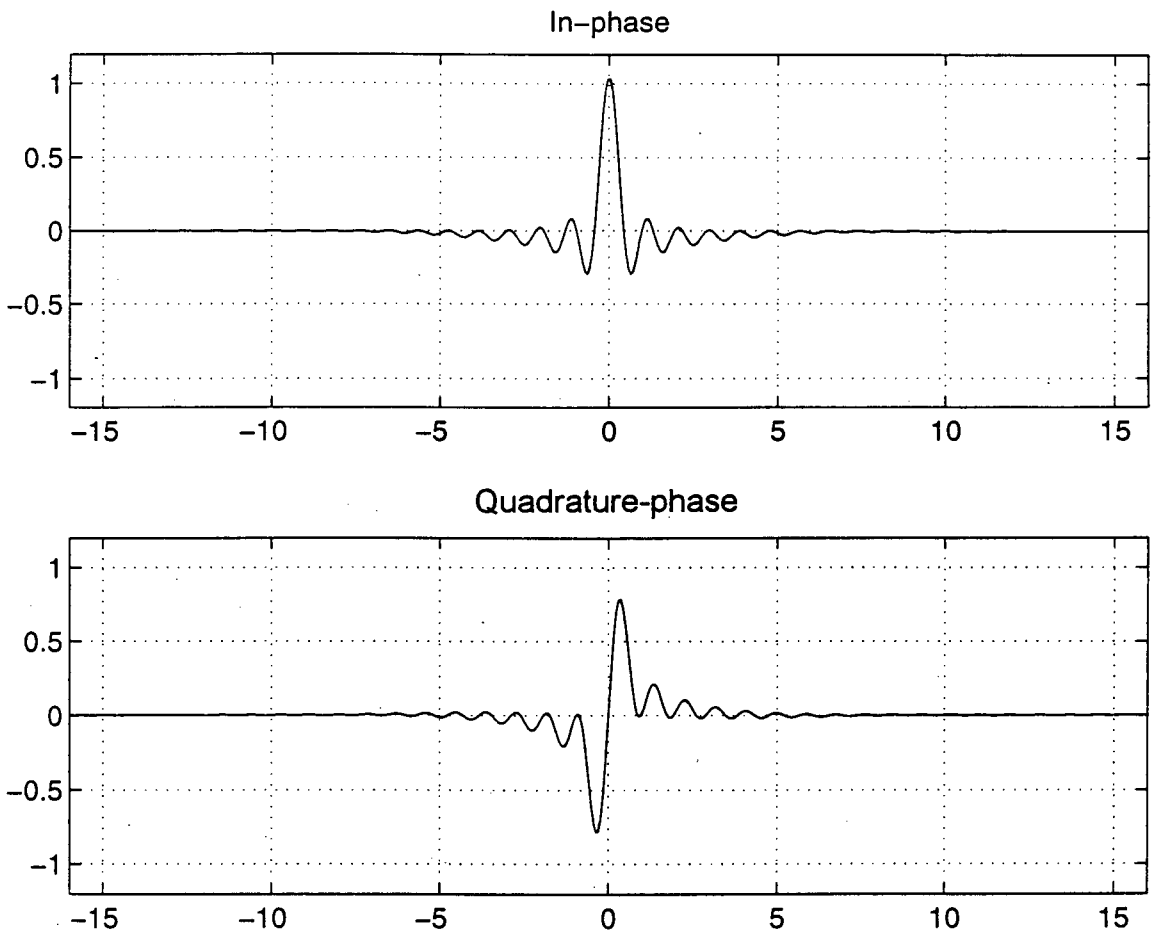


Figure 1.9: Impulse response of shaping filters

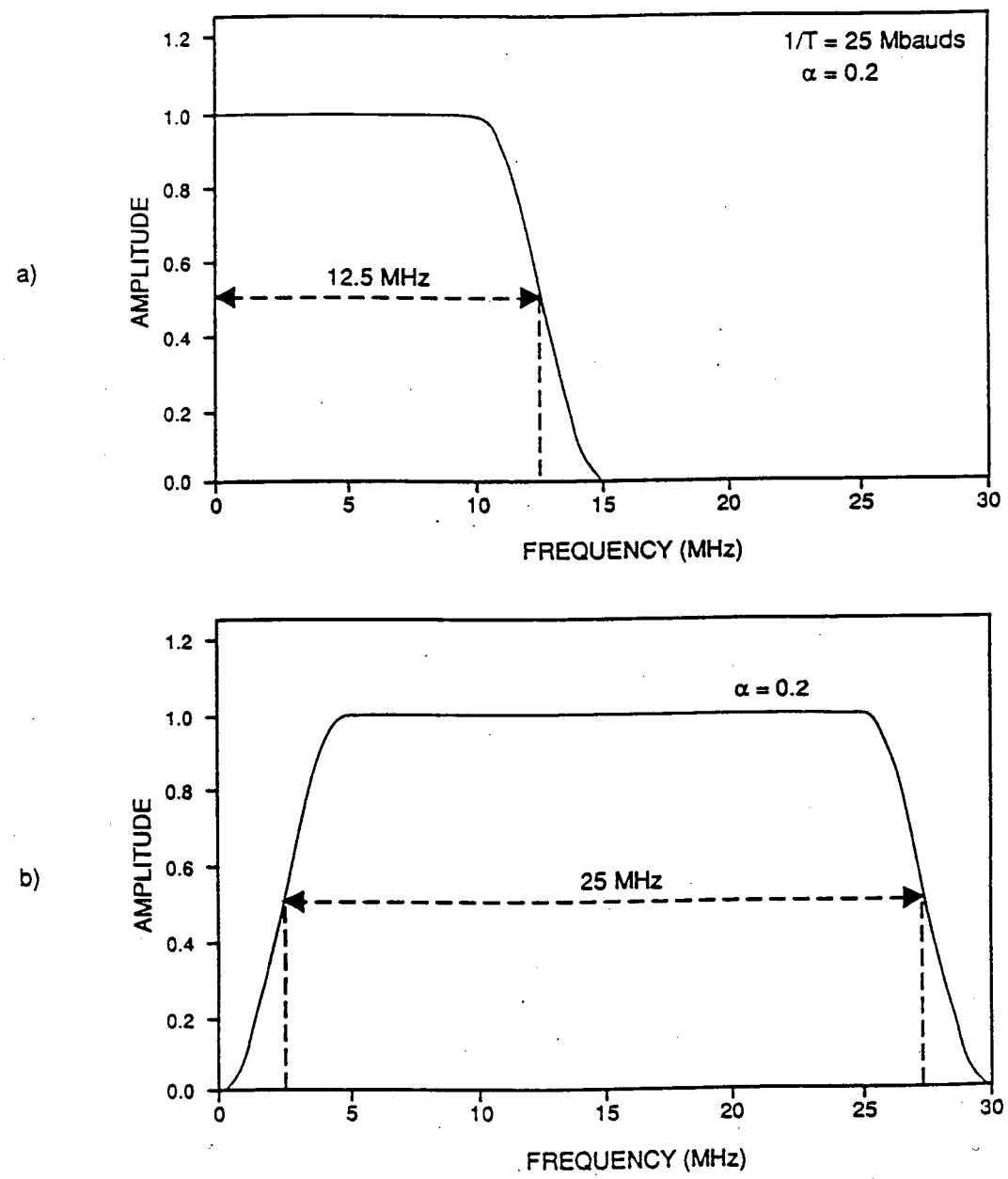


Figure 1.10: Frequency response of shaping filters with different excess bandwidths

At the receiver, the incoming signal $r(t)$ is sampled at the sampling rate $1/T'$. We chose to use the phase-splitting equalizer in our receiver, because it is one of the simplest receiver structures available for the normal mode of operation [30]. Fig. 1.5 shows the structure of the phase-splitting equalizer. In this configuration, an in-phase filter $\mathbf{c}_n$ and a quadrature phase filter $\mathbf{d}_n$ are employed as the two FIR adaptive filters for the equalizer. When an ideal reference is not provided, a decision device is required after the equalizer to determine the transmitted symbols a_n and b_n from the outputs y_n and $\tilde{y}_n$ of the equalizer. Finally, a decoder makes a symbol-to-bit mapping and its output is fed to a descrambler (not shown) whose output is the transmitted bit stream. The phase splitting equalizer will be discussed in more details in subsequent sections.

1.4 The Concept of Blind Equalization

When adaptive equalizers are initially adapted without the help of an ideal reference, they do not always converge. Without an ideal reference, the decisions on the equalizer's outputs have to be made with a decision device, called a slicer, as shown in Fig. 2.1(a). At the beginning of start-up, the equalizer's outputs are corrupted by a lot of intersymbol interference (ISI). In this case, the slicer makes many wrong decisions because of the severely distorted signals. The equalizer cannot converge if the probability of wrong decisions is too high. Blind algorithms have the property of reducing wrong decisions (in some sense), so that equalizers can converge with corrupted data.

Blind algorithms behave differently from the standard least mean square (LMS) algorithm. In the LMS algorithm, the cost function minimizes errors computed between the slicer's input and output or between the equalizer's output and an ideal reference, if available. The cost function minimized by the LMS algorithm is the mean-square error (MSE) defined as

$$CF = J_{min} = E[|E_n|^2(LMS)] = E[|Y_n - A_n|^2] \tag{1.15}$$

where Y_n is the equalizer's complex output, A_n is the slicer's complex output or ideal reference, and $[\cdot]$ means expectation [9]. For blind algorithms, different types of cost functions are utilized. For instance, for one form of the constant modulus algorithm (CMA), the cost function minimizes

the dispersion of the equalizer's complex outputs around a circle

$$CF = J_{ex} = E[e_{r,n}^2(CMA)] = E[(|Y_n|^2 - R^2)^2] \tag{1.16}$$

where the constant R is computed as

$$R = \frac{E[|A_n|^4]}{E[|A_n|^2]} \tag{1.17}$$

and is a statistical function of the symbols a_n and b_n. This type of constant appears in other blind algorithms, but is computed differently. The general principle of a blind start-up procedure is shown in Fig. 1.11. Initially, a blind adaptation algorithm, which minimizes the cost function in (1.16), for example, is used to partially converge the equalizer. We say that the algorithm is used to open the eye diagram, or simply the eye. Once the eye is open, the receiver switches to the standard LMS algorithm, which minimizes the cost function (MSE) in (1.15).

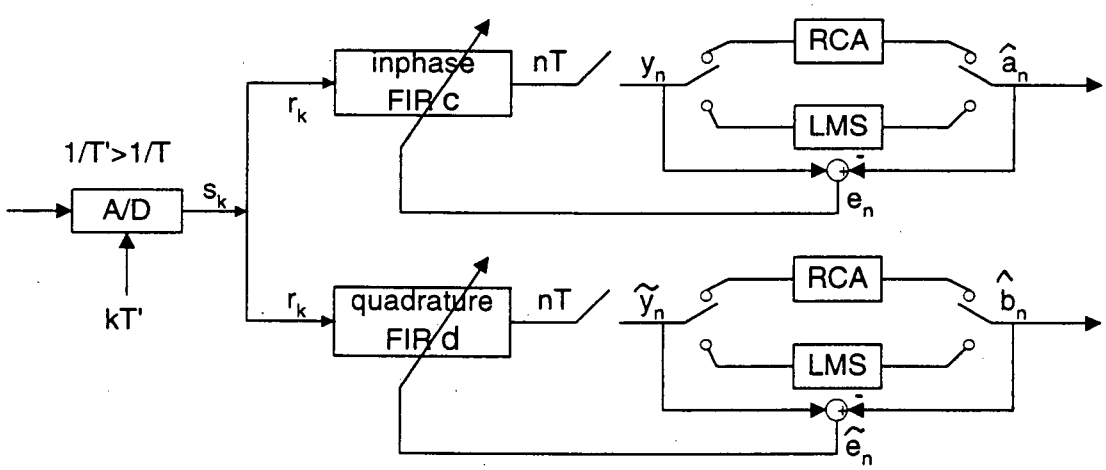


Figure 1.11: Two-step blind equalization

The expressions eye diagram and eye opening were originally coined when signal processing at the receiver was done in the analog world. An analog eye diagram can be observed on an oscilloscope

by superposing all the possible waveforms of the signal in a symbol period when the transmitted data are random. Examples of eye diagrams for two- and eight-level data signal are shown in Fig. 1.12. The expression eye opening refers to the maximum spacing that exists between adjacent signal levels in Fig. 1.12. It should be clear from the figure that the eye diagrams on the right have a larger eye opening than the eye diagrams on the left, and that the latter would be more sensitive to noise than the former. Nowadays, the expressions eye diagram and eye opening are also used to describe a signal constellation and the amount of opening that exists adjacent clusters in the two-dimensional constellation obtained in a digital receiver.

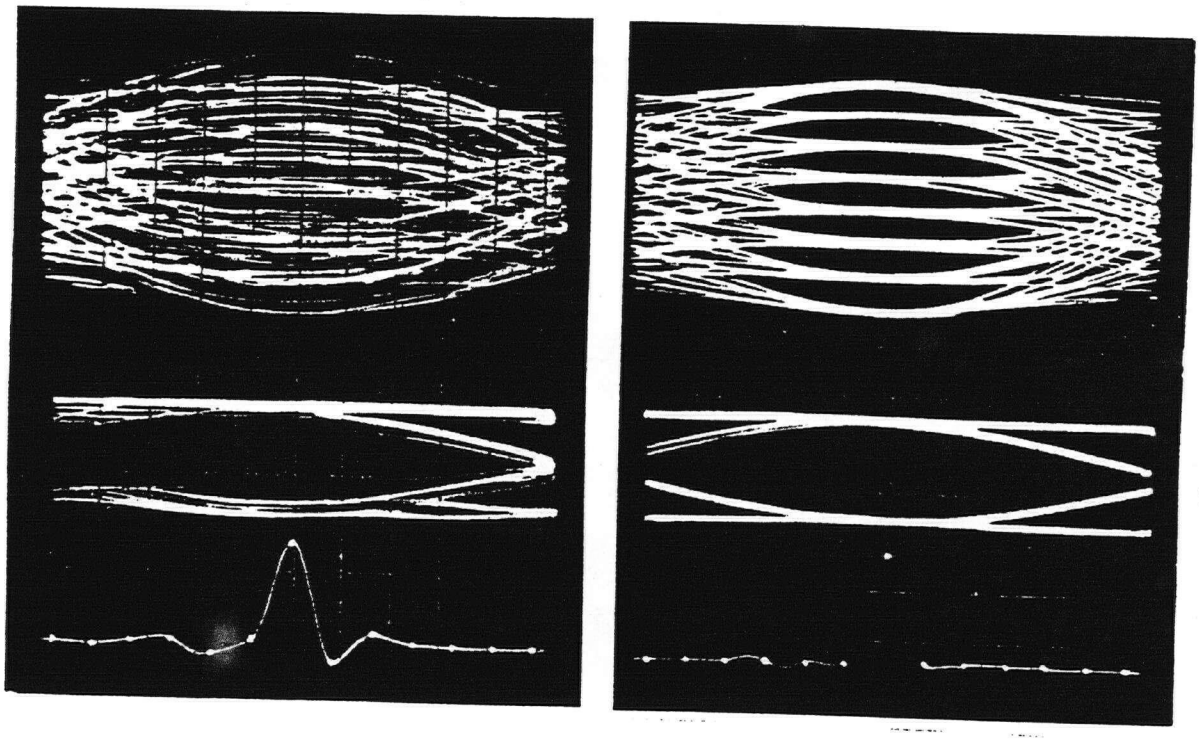


Figure 1.12: Eye diagram and eye opening

1.5 Literature Review

Several blind algorithms have been developed to tackle the problem of blind start-up. The first major contribution to the field of blind equalization was made by Sato in 1975 [24]. The Sato algorithm applies to one-dimensional modulation schemes only. The algorithm was subsequently

generalized to two-dimensional modulation schemes in [8] and [10] and is called RCA in this thesis. Fig. 3.1(a) shows the modulus used by RCA with a 64-CAP signal constellation. During blind start-up, the equalizer minimizes a cost function that is computed with respect to a reduced constellation consisting of four points. The constant R represents the modulus of the four points and is computed from the statistics of the symbols a_n and b_n of the 64-point constellation. More details on RCA will be given in later sections.

Another well-known blind equalization algorithm is called the constant modulus algorithm (CMA) and was first described in detail in [8] and [11] and in some workshop papers by Godard and Benveniste, which are not readily available. This algorithm was then extensively studied in the literature [2], [7], [16] and [19] because of its nice mathematical structure. The algorithm is especially well suited to applications where the receiver has to do both channel equalization and carrier recovery. CMA minimizes the dispersion of the equalizer's output samples with respect to a circle with radius R, as shown in Fig3.1(b). We can achieve more reliable convergence by using CMA than by using RCA. However, as will be shown later, because of the cost function used by CMA, the algorithm does not provide linear phase equalization. This may result in a rotated signal constellation at the output of the equalizer. After achieving convergence with CMA, we have to use other techniques to tune the signal constellation to its final position. This makes the algorithm implementation complicated compared to other algorithms.

Many researchers have investigated the issue of what types of signal statistics should be used for the cost functions of blind algorithms [16] [20] [26]. Most investigations seem to indicate that second-order statistics provide less robust systems than higher order statistics. Our own investigations, reported here, indicate that fourth-order statistics seem to provide the best compromise between performance and cost of implementation.

1.6 Thesis Overview

The thesis is organized as follows. Chapter 2 describes the transceiver used for the ATM LAN standard and other applications. In the same chapter, the problem formulation for the equalizer structure that is being used is given in terms of performance and reliability. Chapter 3 gives a

detailed description of the proposed new MMA algorithm. The algorithm's performance is then compared to the other two major blind algorithms, i.e., RCA and CMA. The following chapter provides an investigation of the effect of the minimum value of the cost functions, which then leads to the definition of the generalized MMA intended for highly bandwidth efficient applications using large signal constellations. In Chapter 5, windowed algorithms for MMA are proposed as an extension of MMA to improve convergence speed. Computer simulations for MMA and its modified versions as well as preliminary experimental results obtained in the laboratory are provided in Chapter 6. Finally, we summarize our results for the blind MMA algorithms, and present suggestions for further work.

Chapter 2

System Configuration

In this thesis, blind equalization algorithms are used for the 155 Mb/s 64-CAP transceiver specified in the ATM LAN standard [1]. However, the same algorithms can equally well be used for other applications, such as the 51 Mb/s 16-CAP transceiver used in the DAVIC specification for FTTC networks [5]. To implement the specified system, we use at the transmitter a two-dimensional transmission scheme with two parallel shaping filters whose impulse responses form a Hilbert pair. At the receiver, several equalizer structures can be used for the two-dimensional CAP transceiver. Among those structures, we have chosen the phase-splitting filter configuration to perform blind equalization. In the next section, we give a description of the transmitter and receiver structures. Then we formulate the problems associated with the chosen equalizer structure.

2.1 Linear Equalizer Structures

At a receiver, several equalizer structures can be used for the two-dimensional CAP transceiver. Three commonly used structures are the two-filter (or phase-splitting) structure shown in Fig. 2.1(a), the cross-coupled structure shown in Fig. 2.2(a), and the four-filter structure shown in Fig. 2.2(b). The use of each structure has its advantages and disadvantages. Decisions have to be made depending on the application. The use of either the cross-coupled filter structure or the four-filter structure generally results in stable blind convergence. However, these equalizers do not necessarily provide the best steady-state performance for a given amount of complexity. By using the two-

filter structure, the equalizer does not necessarily provide stable initial blind convergence, However, this structure gives the best steady-state performance and the simplest implementation among the three structures. Based on the above considerations, we have chosen the two-filter structure for the 64-CAP ATM LAN standard application because it provides the best performance in steady-state operation with the least amount of complexity. The price one has to pay by using the phase-splitting equalizer is that it is difficult to converge blindly at start-up with a high degree of reliability. The new techniques proposed in this thesis and in related work solve this problem [29] [30].

The two-filter structure is shown in Fig. 2.1(a), and details of the structure are given in Fig. 2.1(b). Such a filter is also called a phase-splitting filter because the two filters converge to in-phase and quadrature phase filters $\mathbf{c}_n$ and $\mathbf{d}_n$, respectively, which have phase characteristics that differ by 90 degrees. At the input of the receiver in Fig. 2.1(b) the received signal $r(t)$ can be written as

$$r(t) = \sum_n [a_n s(t - nT) - b_n \tilde{s}(t - nT)] + \xi(t) \tag{2.1}$$

where $1/T$ is the symbol rate, $\xi(t)$ is additive noise introduced in the channel, and $s(t)$ and $\tilde{s}(t)$ are overall impulse responses of the channel and transmitter shaping filters. These impulse responses can be written as

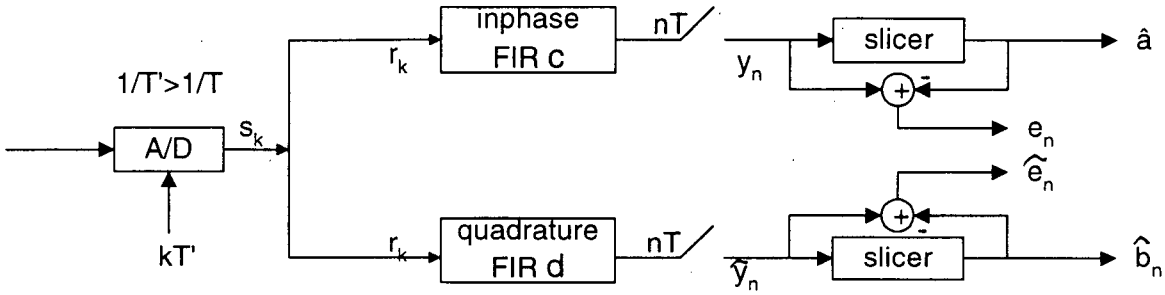
$$s(t) = p(t) \otimes h(t) \qquad \tilde{s}(t) = \tilde{p}(t) \otimes h(t) \tag{2.2}$$

where $p(t)$ and $\tilde{p}(t)$ are the impulse responses of the in-phase and quadrature shaping filters at the transmitter and $h(t)$ is the impulse response of the channel. The output of the fractionally spaced linear equalizer is computed at the symbol rate $1/T$. For the applications considered here, the sampling rate $1/T'$ of the A/D is typically chosen to be three or four times higher than the symbol rate $1/T$, so that $T/T' = i$, where i is a positive integer. The reason for choosing $T/T' = 3$ *or* 4 is to guarantee two desirable features:

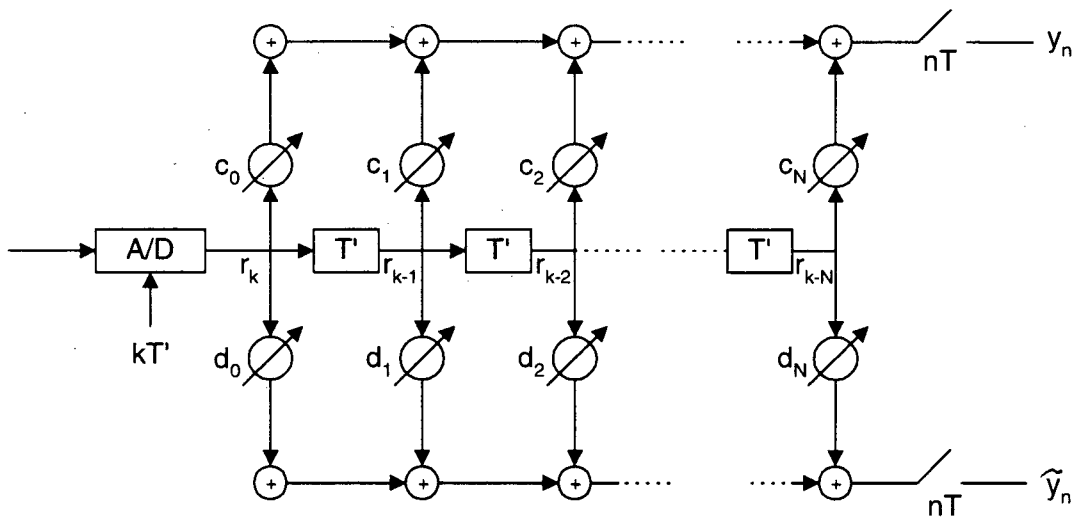
1) Satisfy Nyquist's sampling theorem, i.e. the sampling rate has to be at least twice the highest frequency component of the analog signal.

2) Keep the sampling rate as low as possible, without violating 1) so that the cost of the VLSI implementation of the A/D and D/A is as low as possible.

As seen in Fig. 2.1(a), the received signal $r(t)$ is immediately fed to two adaptive filters $\mathbf{c}$ and $\mathbf{d}$. The two filters share the same tapped delay line which stores sequences of successive A/D samples.

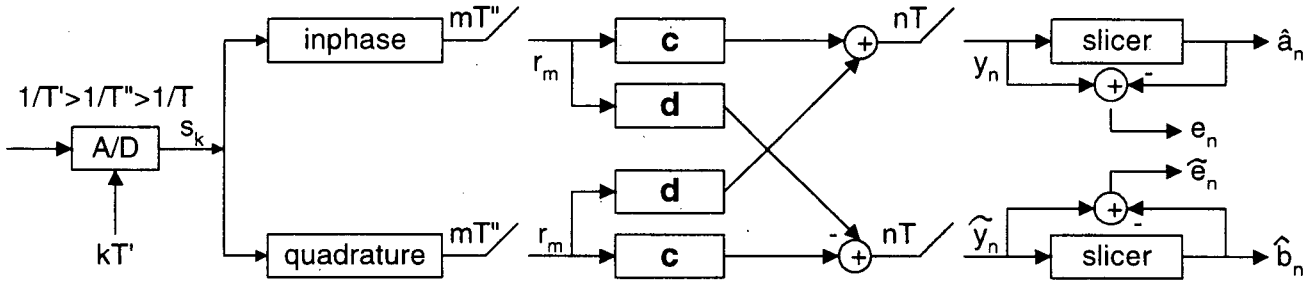


a. Phase-Splitting FSLE

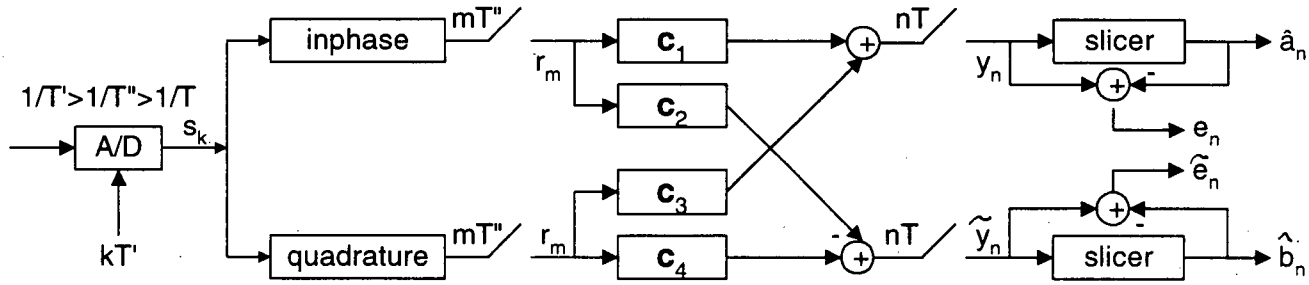


b. Details of Adaptive Filters

Figure 2.1: Block diagram and details of phase-splitting FSLE



a. Cross-Coupled FSLE



b. Four-Filter FSLE

Figure 2.2: Block diagrams of complex FSLEs

The following definitions are made for the vectors $\mathbf{r}(nT)$, $\mathbf{c}(nT)$ and $\mathbf{d}(nT)$

$$\mathbf{r}_n^T = [r_k, r_{k-1}, \ldots, r_{k-N}] = \text{vector of A/D samples in delay line} \tag{2.3}$$

$$\mathbf{c}_n^T = [c_0, c_1, \ldots, c_N] = \text{vector of in} - \text{phase tap coefficients} \tag{2.4}$$

$$\mathbf{d}_n^T = [d_0, d_1, \ldots, d_N] = \text{vector of quadrature tap coefficients} \tag{2.5}$$

where the superscript T denotes vector transpose, and the subscript n is a short notation for the symbol period nT. For the 64-CAP ATM standard, we use $i = 3$, so that we have $k = 3 * n$. The outputs y_n and $\tilde{y}_n$ of the equalizer are computed at the symbol rate $1/T$

$$y_n = \sum_{m=0}^{N} c_m r((k-m)T/i) \qquad \tilde{y}_n = \sum_{m=0}^{N} d_m r((k-m)T/i) \tag{2.6}$$

The equalizer's outputs can also be expressed in vector form

$$y_n = \mathbf{c}_n^T \mathbf{r}_n \qquad \tilde{y}_n = \mathbf{d}_n^T \mathbf{r}_n \qquad Y_n = \mathbf{C}_n^T \mathbf{r}_n \tag{2.7}$$

where we have defined the complex quantities

$$Y_n = y_n + j\tilde{y}_n \qquad \mathbf{C}_n = \mathbf{c}_n + j\mathbf{d}_n \tag{2.8}$$

Note that the tap coefficients are initialized with shaping filters. In steady-state operation, the slicers are used to make decisions $\hat{a}_n$ and $\hat{b}_n$ on the received samples, as shown in Fig. 2.1(a). The equalizer's errors e_n and $\tilde{e}_n$ are computed at the slicers as

$$e_n = y_n - \hat{a}_n = \mathbf{c}_n^T \mathbf{r}_n - \hat{a}_n \qquad \tilde{e}_n = \tilde{y}_n - \hat{b}_n = \mathbf{d}_n^T \mathbf{r}_n - \hat{b}_n \tag{2.9}$$

The tap coefficients of the filters are then updated by using the LMS algorithm. The tap updating algorithms for the two filters are given by

$$\mathbf{c}_{n+1} = \mathbf{c}_n - \mu e_n \mathbf{r}_n \tag{2.10}$$

and

$$\mathbf{d}_{n+1} = \mathbf{d}_n - \mu \tilde{e}_n \mathbf{r}_n \tag{2.11}$$

where μ is a small number called step size. The above equations also provide the generic form for all the tap updating algorithms discussed in this thesis, except that, for blind equalization, e_n and $\tilde{e}_n$ are not computed according to (2.9).

As pointed out in Chapter 1, for a blind start-up we need at least two different tap updating algorithms. First, the eye is opened with a blind algorithm, and then the tap updating algorithms in (2.10) and (2.11) are used to achieve steady-state convergence.

2.2 Problem Formulation

Fig. 1.5 shows a typical communication link incorporating a CAP transceiver. The transmitter, on the top, transmits a signal that represents a block of bits, is mapped to symbols and send out thought shaping filters. The signal is passed through a channel with transfer function $H(f)$. At the output of the channel, some noise $\xi(t)$ is added to the CAP signal and the result appears at the input of the receiver.

The channel can severely distort the CAP signal as shown in Fig. 2.3. Curve 1, in this figure, shows the spectrum of the CAP signal at the output of the transmitter. Curve 2 shows the spectrum of the same signal at the output of the channel, which is assumed to be a 100m unshielded twisted pair category 3 cable. Notice that the signal has been severely attenuated and distorted at higher frequencies after passing through the channel.

For the ATM CAP application, the dominant component of the noise $\xi(t)$ in Fig. 1.5 is near-end crosstalk (NEXT). The power of this type of noise increases by 15 dB per decade with frequency. NEXT will not be discussed any further here because it does not have a significant effect on blind equalization, for reasons which will be discussed later. NEXT modeling is discussed in detail in Refs. [18], [19] and [28].

The purpose of the receiver in Fig. 1.5 is to compensate for the distortion introduced by the channel and to minimize the effects of noise. In this thesis, only linear receivers will be considered. A linear receiver performs linear filtering operations only, and does not perform any nonlinear operations. It is a well-known result that the optimum linear receiver consists of a matched filter followed by a symbol-spaced FIR filter, as shown in Fig. 3.7 [9] [21]. The matched filter maximizes the signal-to-noise ratio (SNR) at the input of the baud sampler, and the FIR filter removes the ISI introduced by the channel and matched filter.

It can be shown that the receiver structure in Fig. 1.5 implements digitally and adaptively the

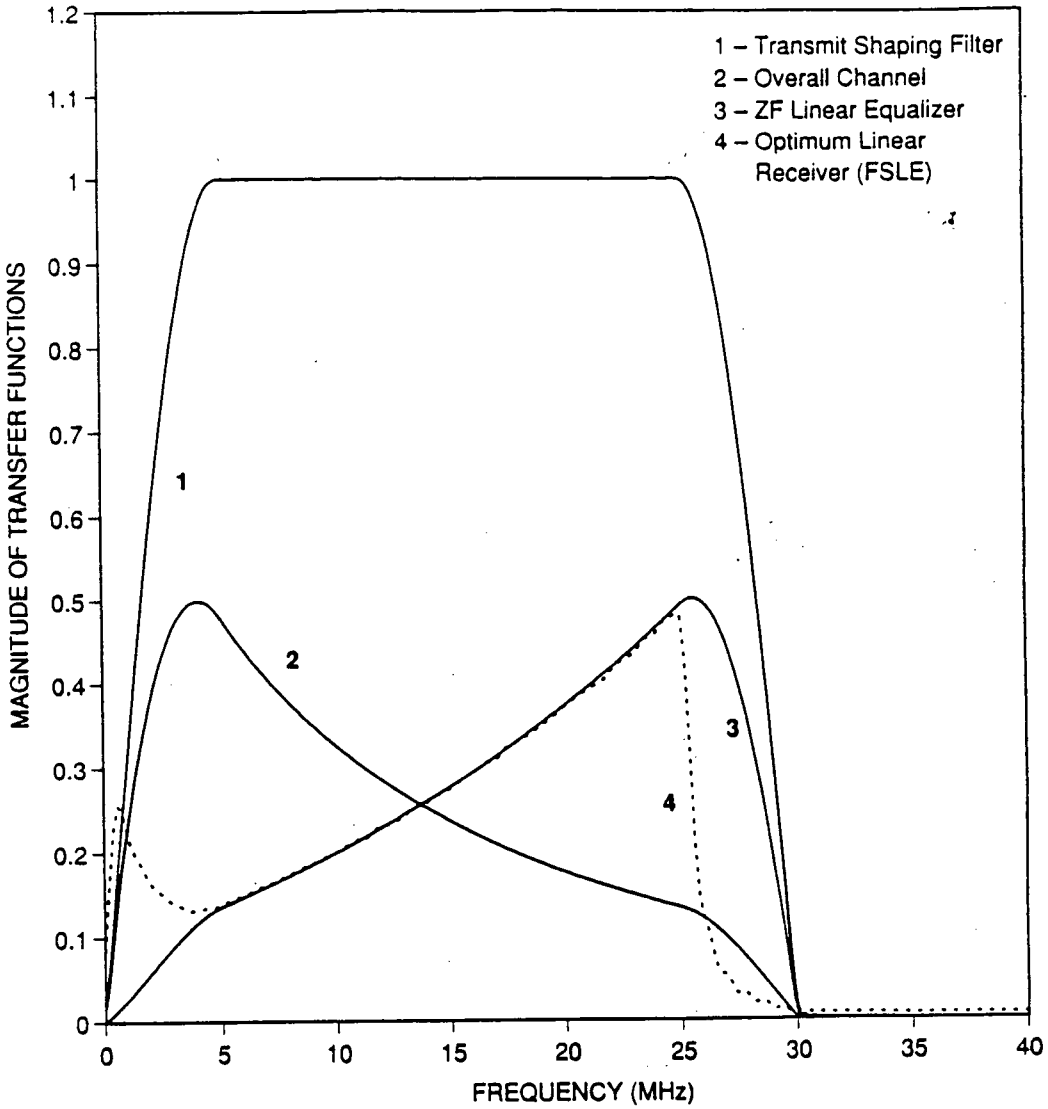


Figure 2.3: Various transfer functions obtained with CAP transceiver

optimum linear receiver for CAP signals. This type of receiver structure is called a fractionally spaced linear equalizer (FSLE) [22]. The magnitude of the transfer function synthesized by the two adaptive filters in Fig. 1.5 is shown as Curve 4 in Fig. 2.3. Curve 3 in this figure corresponds to the transfer function of an equalizer that compensates for the channel distortion only, and does nothing against noise. This is the transfer function synthesized by the so-called zero-forcing linear equalizer, which simply "inverts" the channel. Curve 4 for the optimum linear receiver was obtained in the presence of NEXT. This type of noise has more energy at higher frequencies than at lower frequencies, which explains the shape of the transfer function synthesized by the FSLE. This transfer function goes to zero in the upper roll-off region where the noise is large and the signal is weak, and it amplifies signals in the lower roll-off region, where the signal is strong and the noise is small. In the flat region of the transmitted spectrum, the FSLE has essentially the same transfer function as the zero-forcing equalizer. Before an equalizer has initially converged, it has to deal with a signal which has been severely distorted by the channel, as shown in Fig. 2.3. The signal constellation at the output of the equalizer in Fig. 1.5 before convergence is started is shown in Fig. 2.4. This constellation has nothing is common with the 64-point constellation used at the transmitter (see Fig. 1.6). It should be clear from Fig. 2.4 that an algorithm, such as the decision directed LMS algorithm, which makes a decision on which symbol has been transmitted and then computes an error with respect to this decision, has no chance to be successful in converging. This is why other algorithms, better suited to blind start-up, have to be utilized.

The main challenge in the implementation of blind start-up is to provide reliable initial convergence. For all the equalizer structures, this means that a good initial eye opening of the signal constellation can be achieved. In addition, for the phase-splitting filter structure, this also means that the equalizer does not converge to wrong solutions. In this thesis we concentrate on the former issue. The issue of wrong solutions for the phase-splitting equalizer is discussed in detail in related work [30].

Initial blind convergence is affected by two factors. One factor is the signal distortion caused by large intersymbol interference. The other factor is the residual value of the cost function after convergence. Both effects grow with increasing m, where m denotes the number of symbol levels.

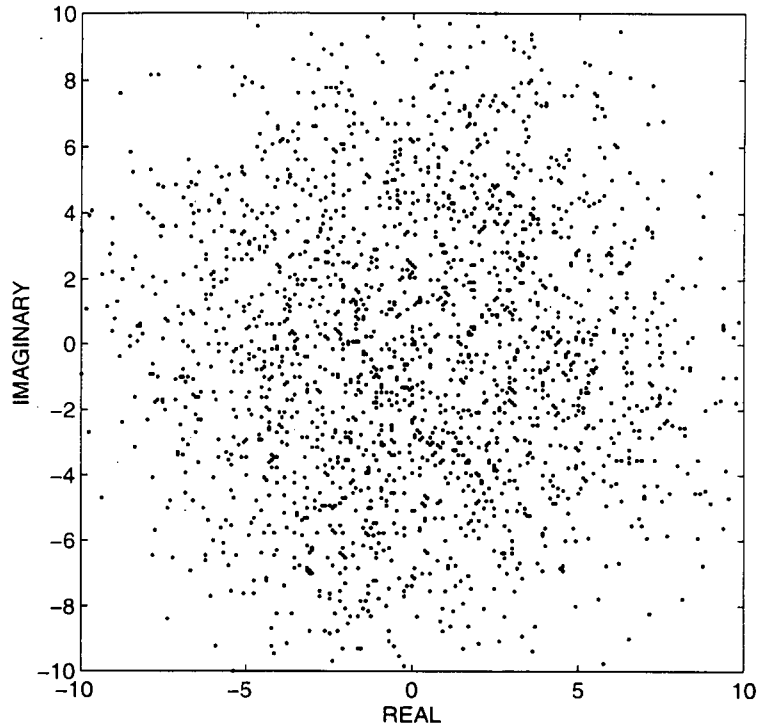


Figure 2.4: Unequalized signal constellation

Fig. 2.5 shows pulses sent by a 256-CAP transmitter. Severe ISI is introduced when these pulses are distorted and start overlapping after passing through a channel.

The performance analysis of blind algorithms is more complicated than for the LMS algorithm. For the LMS algorithm, the cost function used to update the filter's tap coefficients is the same as that used to measure the mean-square error (MSE), and is given by

$$CF = MSE = E[e_{r,n}^2] = E[(y_n - a_n)^2] \tag{2.12}$$

The cost function in (2.12) converges to zero when $y_n \to a_n$. In practice, due to limitations caused by finite filter length, step sizes, *etc*, and the presence of additive noise in the channel, the cost function in (2.12) is not zero but assumes some small values. The LMS algorithm can provide an optimum solution in terms of MSE. However, this is not necessarily the case for blind algorithms. For example, in one possible implementation of MMA, which is discussed later, the cost function

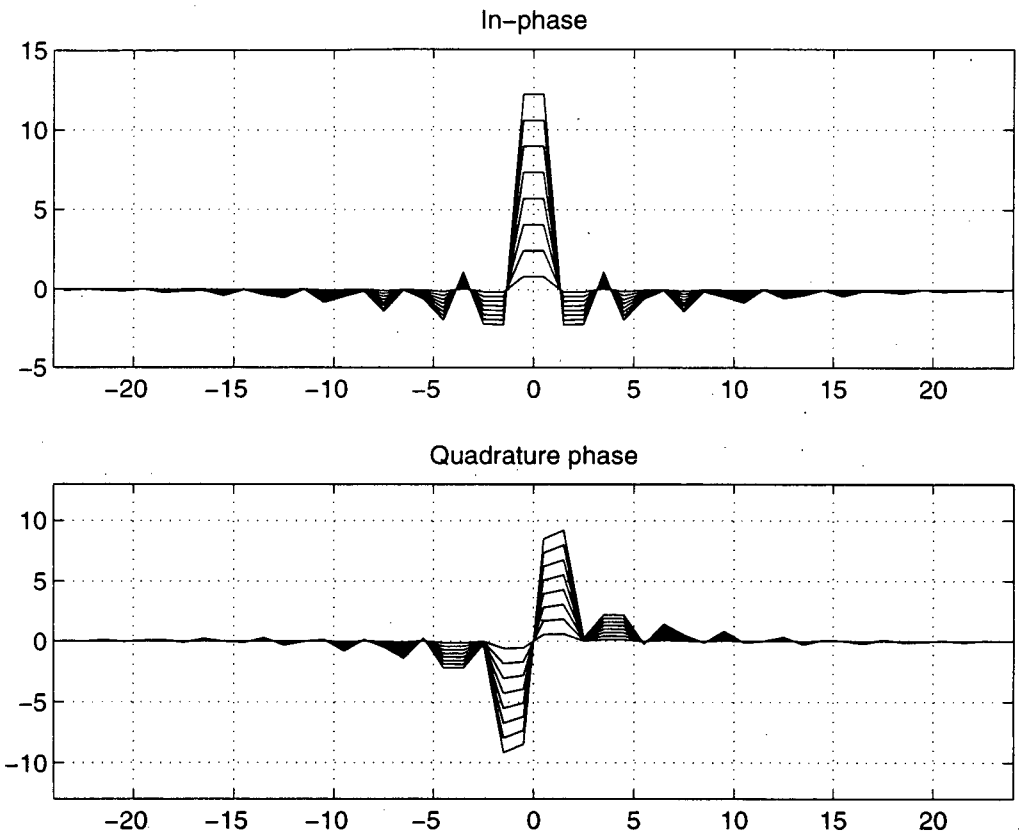


Figure 2.5: Impulse responses of shaping filters for 256-CAP

minimized by the filter's tap updating algorithm for the in-phase tap coefficients is given by

$$CF_y = E[e_{r,n}^2(CF)] = E[(y_n^2 - R^2)^2] \tag{2.13}$$

where y_n is the in-phase output sample of the equalizer. Even when the equalizer converges, i.e. $y_n \to a_n$, we have $CF_y \neq 0$. So that (2.13) becomes

$$CF_a = E[(a_n^2 - R^2)^2] = C \tag{2.14}$$

The stochastic gradient algorithm which minimizes the MSE in (2.12) was previously given in (2.10) and (2.11). The corresponding algorithm which minimizes the MMA cost function in (2.13) will be derived later (see equations (3.29) and (3.30). We repeat here these algorithms as they are used

for the in-phase tap coefficients:

$$\text{MSE}: \quad \mathbf{c}_{n+1} = \mathbf{c}_n - \mu(y_n - a_n)\mathbf{r}_n \tag{2.15}$$

$$\text{MMA}: \quad \mathbf{c}_{n+1} = \mathbf{c}_n - \mu(y_n^2 - R^2)y_n\mathbf{r}_n \tag{2.16}$$

During final convergence, i.e., when $y_n \to a_n$, the correction term for the LMS algorithm in (2.15) goes to zero. As a result, the tap coefficients settle to their optimum values with very little tap fluctuations. The same is not true for the tap updating algorithm in (2.16). When $y_n \to a_n$, the correction term does not go to zero. As a result, the tap coefficients keep fluctuating around their optimum values. This creates tap fluctuation noise at the output of the equalizer, and does not allow the MSE in (2.12) to go to zero. The only way to have the MSE go to zero would be to use a step size that goes to zero, i.e. $\mu \to 0$ in (2.16). However, this may not be possible in a finite-precision environment, and would not be desirable even if enough precision were available, because the convergence time would be too long.

As the number m of symbol levels increases, the tap fluctuation noise introduced by a blind updating algorithm becomes more and more significant when compared to the spacing between adjacent symbols in the signal constellation. As a result, as m increases, it becomes more and more difficult to open the eye. Our investigation has shown that there is a direct correlation between the ease with which the eye can be open and the minimum (residual) value of the cost function that is being minimized by the blind algorithm. The larger this minimum value, the more difficult it becomes to open the eye.

A detailed study of the effect of the residual values of cost functions is given in Section 4.1 and Appendix C, or in [30]. In Table 2.1, we list computed values for several CAP systems. We see that the residual values of the cost functions increase with increasing m. The residual value is about 14 dB for 16-CAP, and increases to 52 dB for 1024-CAP. The above result has led to the proposal of the modified MMA algorithms, which are presented in Chapters 4 and 5. The cost functions of these modified algorithms have residual values which are smaller than the residual value of the cost function of the basic MMA algorithm. As a result, the modified MMA algorithms can be used with denser signal constellations and/or provide faster initial blind convergence.

CF/CAP	4-CAP	16-CAP	64-CAP	256-CAP	1024-CAP
m	1	2	4	8	16
CF_{an}	0	14.2	27.7	40	52

Table 2.1: Minimum values of the MMA cost function

Chapter 3

MultiModulus Algorithm (MMA)

The main challenge in the implementation of blind start-up is to provide a reliable convergence. Reliable convergence of the equalizer includes two features. One feature is the opening of the eye diagram, and the other one is avoiding wrong solutions. In the overview of blind algorithms, we will see that both RCA and CMA cannot guarantee both features at the same time. To meet the requirements for blind equalization, we propose in this chapter an improved algorithm, called MMA. Enhanced versions of this algorithm will be discussed in the next two chapters. The major attribute of MMA is the usage of multiple moduli, which provides more flexibility and leads to an improved convergence. We also show that MMA is a blind algorithm which can remove all ISI. At the end of this chapter, MMA is analyzed and compared to RCA and MMA.

3.1 Overview of Blind Algorithms

RCA and CMA are two different blind algorithms. However, during cost function minimization, both of them employ constants which relate to the statistics of the symbols a_n and b_n. The principle of the two algorithms is illustrated in Fig. 3.1(a) and (b). For RCA, the cost function is defined with respect to four points, as shown in Fig. 3.1(a), and for CMA, the cost function is defined with respect to a circle, as shown in Fig. 3.1(b). The values of the constants which define the four points for RCA and the circle for CMA are functions of the statistics of the symbols a_n and b_n.

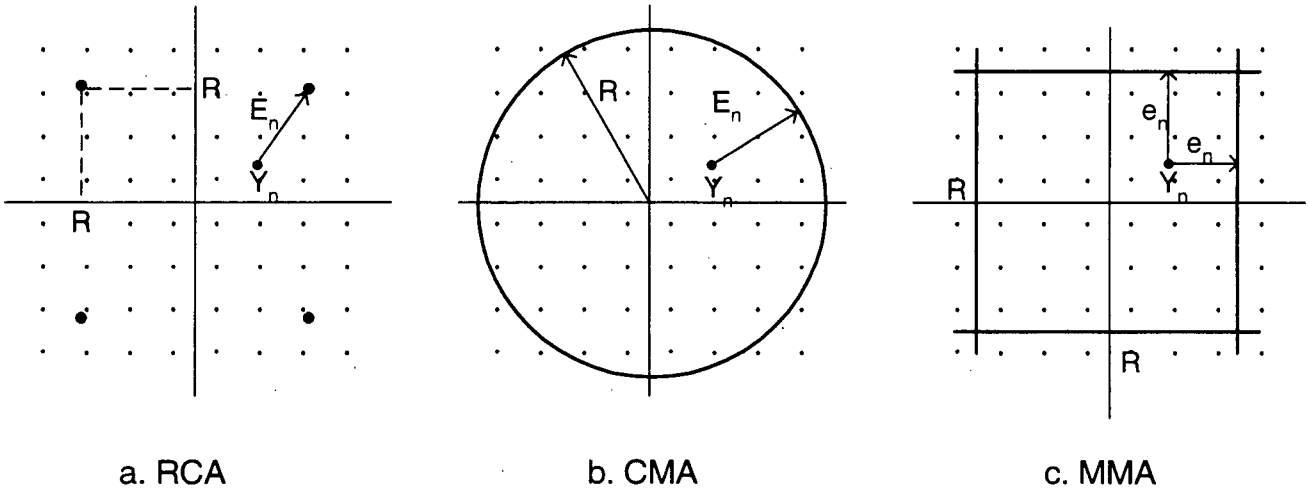


Figure 3.1: Interpretation of the cost function used by three differrent blind equalization

3.1.1 Reduced Constellation Algorithm (RCA)

The first algorithm considered in the overview of blind algorithms is RCA, which has the simplest implementation [9]. Before discussing RCA we briefly revisit the LMS algorithm, which is used in the steady-state mode of operation. With the LMS algorithm, the cost function that is minimized is the MSE, where the error is computed by subtracting the outputs of the slicer from its inputs. The MSE is expressed as

$$MSE = E[|E_n|^2] = E[|Y_n - \hat{A}_n|^2] \tag{3.1}$$

where Y_n and $\hat{A}_n$ represent inputs and outputs of the slicer respectively. In (3.1), the complex quantities are defined in the following way

$$Y_n = y_n + j\tilde{y}_n \quad \hat{A}_n = \hat{a}_n + j\hat{b}_n \quad E_n = e_n + j\tilde{e}_n \tag{3.2}$$

$$E_n = Y_n - \hat{A}_n \quad e_n = y_n - \hat{a}_n \quad \tilde{e}_n = \tilde{y}_n - \hat{b}_n \tag{3.3}$$

Without ideal reference, many wrong decisions would be made at the slicers during initial convergence, and the equalizer would usually not converge with the LMS algorithm during blind start-up.

In this case, a blind algorithm is required to solve the problem of making wrong decisions. The key element in blind algorithms is the use of the statistics of the symbols A_n. Instead of using individual symbols A_n as reference, a blind equalizer uses correction terms, which are derived with respect to constants whose values are functions of the statistics of the symbols A_n. By applying such a technique, the probability of wrong decisions can be significantly reduced. This concept is now demonstrated for RCA. With the RCA algorithm, the error used in the tap updating algorithm is derived with respect to a signal constellation that has a smaller number of points than the received constellation. As illustrated in Fig. 3.1 (a), it is assumed that the original signal constellation uses 64 complex symbols. For the RCA algorithm, the reduced constellation typically consists of four signal points only. It should be noted that the RCA algorithm requires the use of a decision device to select the closest signal point from the reduced constellation. The error between the received sample Y_n and the closest signal point $\hat{A}_{r,n}$ of the reduced constellation is the complex number

$$E_{r,n} = Y_n - \hat{A}_{r,n} = Y_n - R * sgn(Y_n) \tag{3.4}$$

where the constant R is a statistical function of the symbols A_n, $sgn(\cdot)$ is the signum function, and the subscript r indicates that the error is derived with respect to a reduced constellation. Note that the expression $Y_n - Rsgn(Y_n)$ corresponds to the case where the reduced constellation consists of four points. The reduced constellation algorithm minimizes the following cost function

$$CF = E[|Y_n - \hat{A}_{r,n}|^2] = E[|Y_n - R * \mathrm{sgn}(Y_n)|^2] \tag{3.5}$$

A derivation for the value of the constant R can be found in Appendix B. The main steps for computing this constant are briefly repeated here. The constant R is computed under the condition that the channel is perfectly equalized, that is, $y_n \to a_n$ and $\tilde{y}_n \to b_n$. Using (2.10) and (2.11) in (3.5), we get the following expression for the gradient of the cost function in (3.5) with respect to the complex tap vector $\mathbf{C}_n$

$$\nabla_{\mathbf{c}}(CF) = 2E[(Y_n - Rsgn(Y_n))\mathbf{r}_n] \tag{3.6}$$

Setting $\nabla_{\mathbf{c}}$ to zero and assuming perfect equalization, we get the following value for R

$$R = \frac{E[|A_n|^2]}{E[|a_n|] + E[|b_n|]} \tag{3.7}$$

Note that the constant R is a statistical function of the symbols A_n and that it takes different values for different constellations.

Now, consider the phase-splitting equalizer structure shown in Fig. 2.1. Using equations (2.7) and (3.5), the following equations result

$$e_{r,n} = y_n - \hat{a}_{r,n} = \mathbf{c}_n^T \mathbf{r}_n - Rsgn(y_n) \quad \tilde{e}_{r,n} = \tilde{y}_n - \hat{b}_{r,n} = \mathbf{d}_n^T \mathbf{r}_n - Rsgn(\tilde{y}_n) \tag{3.8}$$

The gradients of the cost function in equation (3.5) with respect to the tap vectors $\mathbf{c}_n$ and $\mathbf{d}_n$ are equal to

$$\nabla_{\mathbf{c}}(CF) = 2E[e_{r,n}\mathbf{r}_n] \tag{3.9}$$

$$\nabla_{\mathbf{d}}(CF) = 2E[\tilde{e}_{r,n}\mathbf{r}_n] \tag{3.10}$$

The nonaveraged gradients in (3.9) and (3.10) can be used in a stochastic gradient algorithm to adapt the tap coefficients of the equalizer, so that the following tap updating algorithms result

$$\mathbf{c}_{n+1} = \mathbf{c}_n - \mu e_{r,n}\mathbf{r}_n = \mathbf{c}_n - \mu[y_n - Rsgn(y_n)]\mathbf{r}_n \tag{3.11}$$

$$\mathbf{d}_{n+1} = \mathbf{d}_n - \mu \tilde{e}_{r,n}\mathbf{r}_n = \mathbf{d}_n - \mu[\tilde{y}_n - Rsgn(\tilde{y}_n)]\mathbf{r}_n \tag{3.12}$$

where $\mathbf{r}$ is initialized with zeros and $\mathbf{c}$ and $\mathbf{d}$ are initialized with values corresponding to the in-phase and quadrature shaping filters. The above tap updating algorithms have been derived for the phase-splitting equalizer structure. Other commonly used equalizer structures are the cross-coupled and the four-filter FSLE structures shown in Fig. 2.2. A detailed study of the implementation of RCA for these structures can be found in [30].

For RCA, the cost function minimization is done with respect to a reduced signal constellation that consists of four-points. Because of the usage of the constant R, the probability of making wrong decisions (in a statistical sense) is significantly reduced, and this leads to initial equalizer convergence. After achieving initial convergence with RCA, the equalizer switches to the LMS algorithm to obtain steady-state convergence performance.

One of the problems of RCA when used with the phase-splitting equalizer is that it can easily converge to wrong solutions. The most commonly observed wrong solution is the diagonal solution, which is obtained when the in-phase and quadrature filter synthesize the same transfer function.

The result is that the outputs of the two filters are the same and the signal constellation becomes a diagonal, as shown in Fig. 3.2.

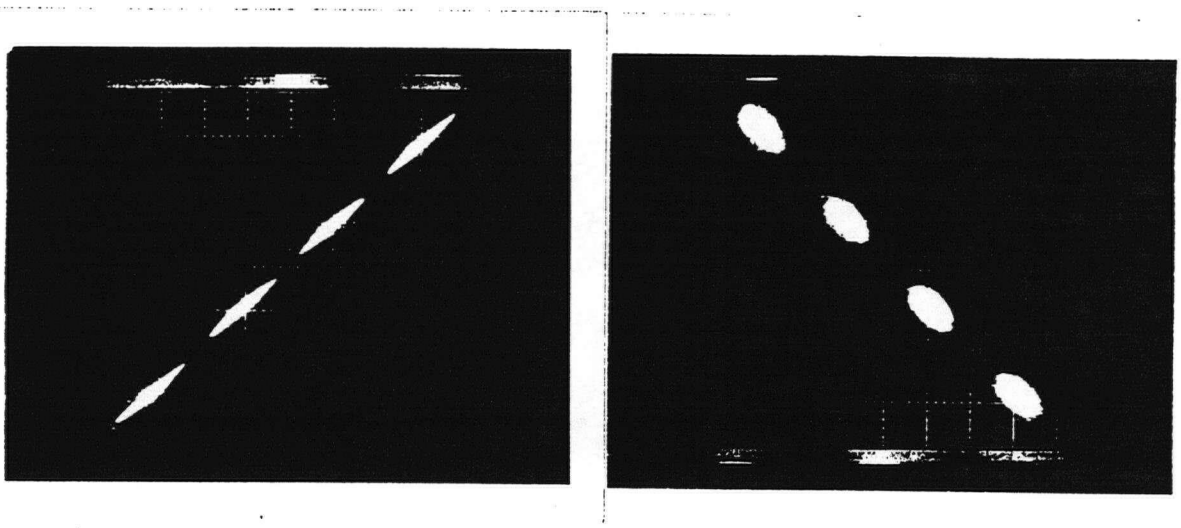


Figure 3.2: Diagonal solution for 64-CAP

3.1.2 Constant Modulus Algorithm (CMA)

This section provides a general overview of the CMA algorithm [9] [11]. The CMA algorithm minimizes the dispersion of the equalizer's output samples Y_n with respect to a circle with radius R. This is graphically illustrated in Fig. 3.1(b). Unlike RCA, which uses second-order statistics only, CMA has a generalized form which uses higher-order statistics of the equalizer's output signals. The CMA algorithm minimizes the following cost function:

$$CF = E[(|Y_n|^L - R^L)^2] \tag{3.13}$$

where L is a positive integer. With CMA, a true two-dimensional cost function is used to minimize the dispersion of the complex outputs Y_n with respect to a two-dimensional contour. A value $L = 2$

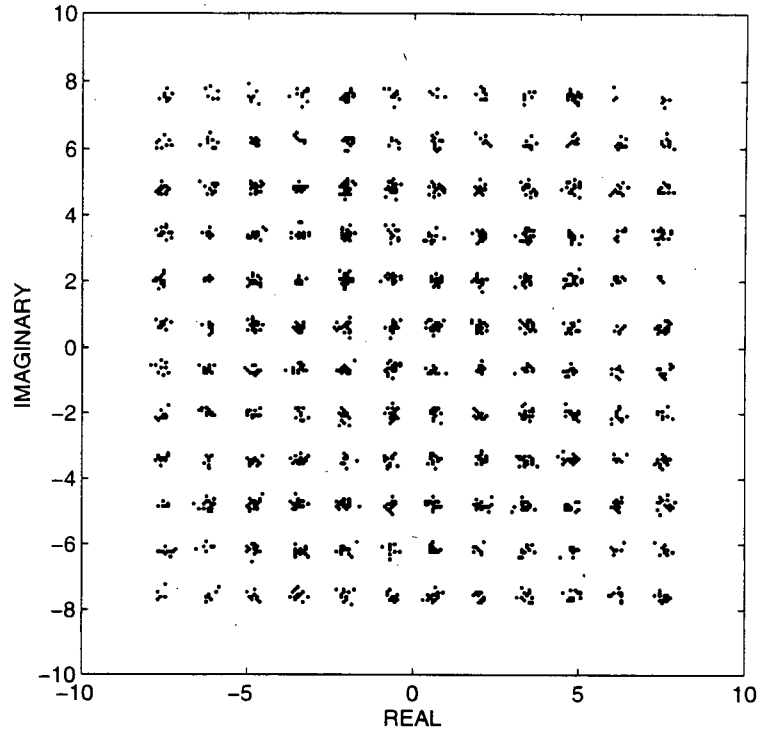


Figure 3.3: 144-point solution for 128-CAP

is commonly used in practice, so that the cost function in (3.13) becomes

$$CF = E[(|Y_n|^2 - R^2)^2] \tag{3.14}$$

Now, consider the phase-splitting equalizer structure in Fig. 2.1. Using (2.7), the gradients of the cost function in (3.14) with respect to the tap vectors $\mathbf{c}_n$ and $\mathbf{d}_n$ are given by

$$\nabla_c(CF) = E[(|Y_n|^2 - R^2)y_n\mathbf{r}_n] \quad \nabla_d(CF) = E[(|Y_n|^2 - R^2)\tilde{y}_n\mathbf{r}_n] \tag{3.15}$$

Setting the gradients to zero and assuming a perfectly equalized channel we get the following value for R^2

$$R^2 = \frac{E[|A_n|^4]}{E[|A_n|^2]} \tag{3.16}$$

This expression holds for the usual case where the statistics of the symbols a_n and b_n are the same. For $L = 2$, we have the following stochastic gradient tap updating algorithms

$$\mathbf{c}_{n+1} = \mathbf{c}_n - \mu(|Y_n|^2 - R^2)y_n\mathbf{r}_n \tag{3.17}$$

$$\mathbf{d}_{n+1} = \mathbf{d}_n - \mu(|Y_n|^2 - R^2)\tilde{y}_n\mathbf{r}_n \tag{3.18}$$

CMA differs from RCA in several respects. RCA only uses knowledge of second-order statistics of the signals, whereas CMA can use higher-order statistics. It is not practical to use statistics which are higher than the fourth-order statistics, because of trade-off considerations between cost and performance. For CMA, the initial dynamic behavior is an increasing function of the order of the statistics being used. However, the computation of the equalizer coefficients quickly becomes unmanageable in (3.17) and (3.18) when L increases. In a digital implementation, a processor based receiver with finite coefficient precision would suffer precision and overflow problems [11]. In this thesis, we always assume the usage of the fourth-order CMA algorithm with $L = 2$ in (3.13). Another difference between the two algorithms is the error contour used for the cost function. The cost function minimization of RCA refers to a four-point constellation, whereas in CMA, the minimization refers to a circle. Even though the cost function of RCA is defined in two dimensions, it can equally well be defined as two separate cost functions in one dimension. However, CMA uses a true two-dimensional cost function, which makes the two channels dependent. Equalization of linear phase offset cannot be done because of the following reasons. Assume that we rotate the complex tap vector $\mathbf{C}_n$ by an angle θ, i.e. $\mathbf{C}_n \rightarrow \mathbf{C}_n e^{j\theta}$. It is easily verified from (2.7) that this will also rotate the complex output sample Y_n of the equalizer by an angle θ, i.e. $Y_n \rightarrow Y_n e^{j\theta}$. Using this rotated output in the CMA cost function in (3.14) we get

$$CF(\theta) = E[(|Y_n e^{j\theta}|^2 - R^2)^2] = E[(|Y_n|^2 - R^2)^2] \tag{3.19}$$

Thus, rotating the complex tap vector $\mathbf{C}_n$ does not change the value of the cost function, including its minimum value. Any rotated version of a tap vector $\mathbf{C}_n$ that minimizes the CMA cost function also provides a minimum of this cost function. As a result, even though CMA can open the eye, it generally will generate a rotated signal constellation at the output of the equalizer, as shown in Fig. 3.4 and Fig. 3.5. This undesirable feature of CMA is a direct consequence of the cost function that is used, and does not depend on the equalizer structure under consideration.

In practical applications where CMA is used, the equalizer is followed by an adaptive rotator, which repositions the signal constellation in the right place. The phase of signal constellation is

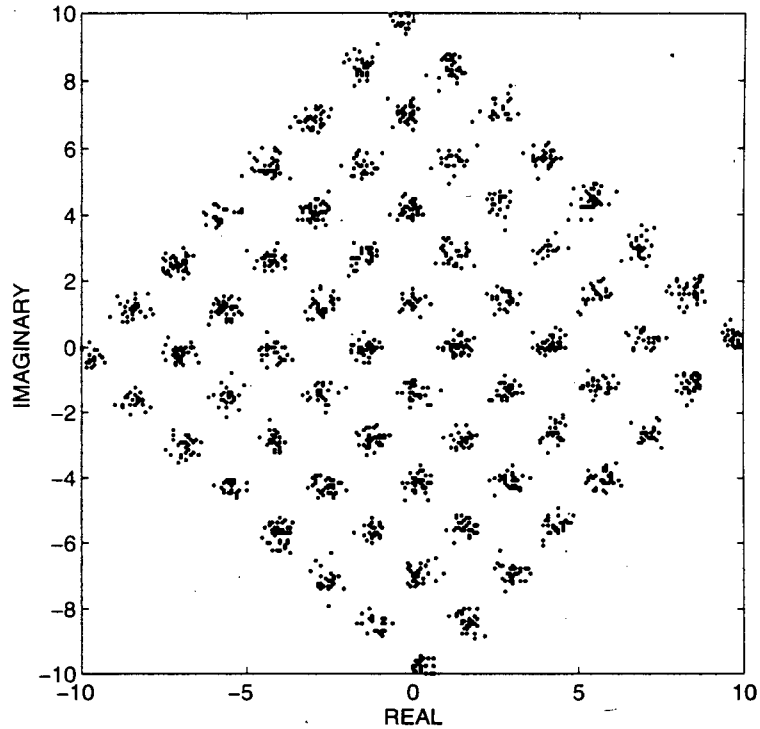


Figure 3.4: 45 degree rotation

computed as

$$\theta = E[\frac{Im\{Y_n * A_n^*\}}{|A_n^2|}] \tag{3.20}$$

Then the gradient descent algorithm can be used to update filter taps

$$\theta_{n+1} = \theta_n - \mu \nabla_\theta \tag{3.21}$$

The rotator must also be used in the steady-state mode of operation, which is undesirable for the applications considered here, because it increases the steady-state complexity and power dissipation of the receiver.

3.2 MMA

Through the overview of blind algorithms, we have seen the problems associated with RCA and CMA. RCA is simple but not very reliable, and especially prone to creating diagonal solutions. The use of CMA can avoid diagonal solutions, but CMA cannot complete an entire equalization and requires additional functions, which makes it more expensive. MMA is proposed as an improved

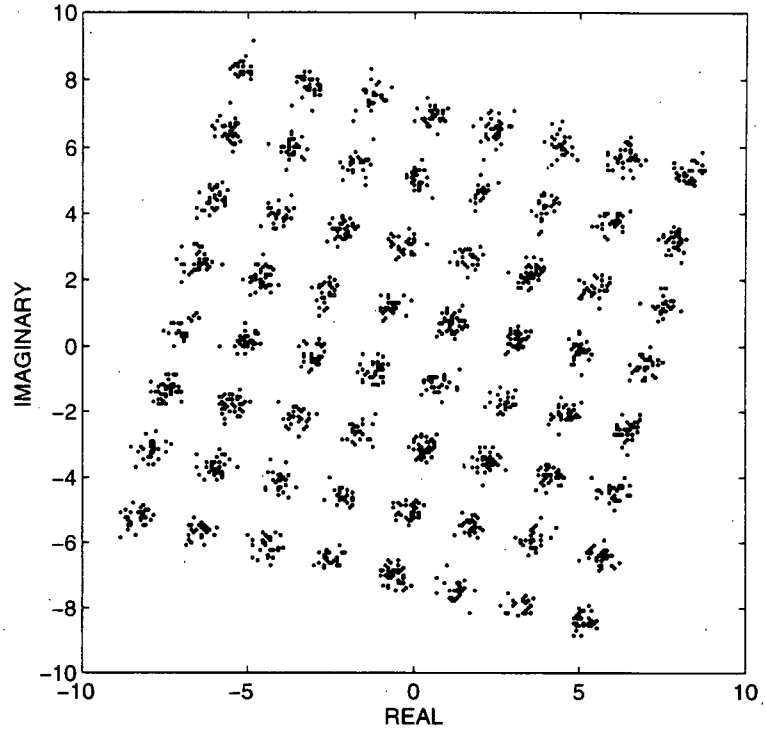


Figure 3.5: 15 degree rotation

blind algorithm to overcome the problems created by RCA and CMA [29]. It is more reliable than RCA and does not require the additional complexity needed by CMA for steady-state operation.

The main feature of MMA is the use of multiple moduli, which makes it very flexible and easily adaptable to nonsquare constellations and to large constellations. The algorithm using multiple moduli for large constellations is called generalized MMA, and this algorithm will be studied in the following chapter. In this section, we study the MMA algorithm's use of multiple moduli for nonsquare constellations. In the following, we first describe the usage of MMA for square constellations such as 64-CAP, and then for nonsquare constellations such as 128-CAP, for which multiple moduli are used to accommodate the nonuniform distribution of equalizer output samples.

3.2.1 Square Constellations

The main feature of MMA is that it minimizes the dispersion of the equalizer's output samples around multiple piecewise linear moduli. All possible usages of moduli can be summarized in one general expression, which is

$$CF = E[(y_n^L - R^L(y_n)^2 + (\tilde{y}_n^L - R^L(\tilde{y}_n)^2] \tag{3.22}$$

where L is a positive integer, and $R(y_n)$ and $R(\tilde{y}_n)$ are output-sample-dependent constants. The values of $R(y_n)$ and $R(\tilde{y}_n)$ are determined by the region in the complex plane in which the equalizer outputs y_n and $\tilde{y}_n$ are located. In comparison to CMA, MMA makes two changes. One change is made by treating the two dimensions separately. The other change is the usage of multiple moduli, that is, R is a function of the distribution of the equalizer's outputs y_n and $\tilde{y}_n$. The two changes make MMA behave differently from CMA.

We first study MMA for square constellations, which is the simplest form of MMA. For square constellations, $R(y_n) = R(\tilde{y}_n) = R = constant$, so that the cost function in (3.22) becomes

$$CF = CF_I + CF_Q = E[(y_n^L - R^L)^2] + E[(\tilde{y}_n^L - R^L)^2] \tag{3.23}$$

Equation (3.23) includes two cost functions. Unlike CMA, this is not a true two-dimensional cost function. Rather, the cost function CF in (3.23) is the sum of two one-dimensional cost functions CF_I and CF_Q. Graphically, the moduli of MMA for square constellations are given by two straight lines along each dimension, as shown in Fig. 3.1(c).

For the phase-splitting equalizer structure shown in Fig. 2.1, the gradients of the cost function in (3.23) with respect to the tap vectors $\mathbf{c}_n$ and $\mathbf{d}_n$ are equal to

$$\nabla_c(CF) = 2L * E[(y_n^L - R^L)^2 |y_n|^{L-2} y_n r_n] \tag{3.24}$$

$$\nabla_d(CF) = 2L * E[(\tilde{y}_n^L - R^L)^2 |\tilde{y}_n|^{L-2} \tilde{y}_n r_n] \tag{3.25}$$

Assuming a perfectly equalized channel, i.e. $y_n \to a_n$ and $\tilde{y}_n \to b_n$, and setting the gradients to zero the following value for the constant R^L is obtained

$$R^L = \frac{E[a_n^{2L}]}{E[|a_n|^L]} = \frac{E[b_n^{2L}]}{E[|b_n|^L]} \tag{3.26}$$

Due to the use of the independent cost functions and because the symbols a_n and b_n are assumed to have the same statistics, the constant R can be derived from either CF_I or CF_Q. For the general form of MMA in (3.23), the following stochastic gradient tap updating algorithms apply

$$\mathbf{c}_{n+1} = \mathbf{c}_n - \mu(y_n^L - R^L)|y_n|^{L-2} y_n \mathbf{r}_n \tag{3.27}$$

$$\mathbf{d}_{n+1} = \mathbf{d}_n - \mu(\tilde{y}_n^L - R^L)|\tilde{y}_n|^{L-2} \tilde{y}_n \mathbf{r}_n \tag{3.28}$$

The best compromise between cost and performance is achieved with $L = 2$, in which case the tap updating algorithms become

$$\mathbf{c}_{n+1} = \mathbf{c}_n - \mu(y_n^2 - R^2)y_n\mathbf{r}_n \tag{3.29}$$

$$\mathbf{d}_{n+1} = \mathbf{d}_n - \mu(\tilde{y}_n^2 - R^2)\tilde{y}_n\mathbf{r}_n \tag{3.30}$$

where the constant R is computed as

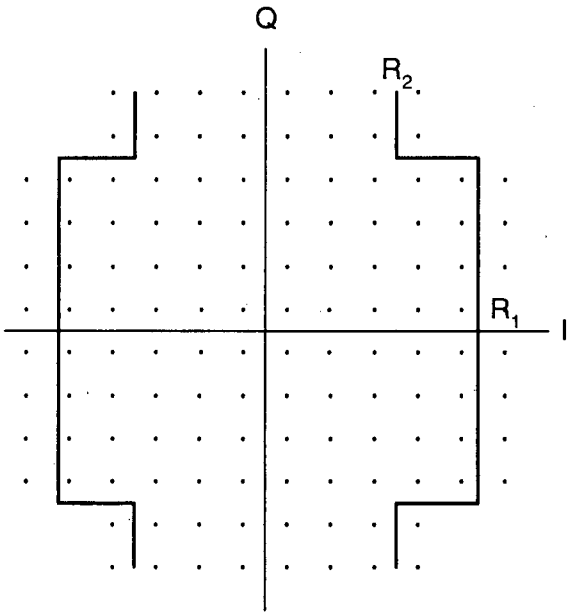
$$R^2 = \frac{E[a_n^4]}{E[a_n^2]} \tag{3.31}$$

In this section, the MMA tap updating algorithms have been derived for the phase-splitting equalizer structure. The corresponding algorithms for the cross-coupled and four-filter FSLE structures can be found in [29].

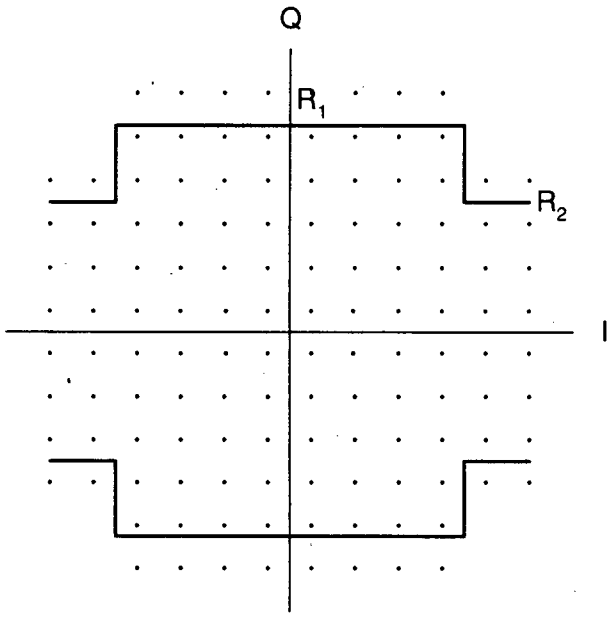
Although CMA and MMA have similar algorithmic structures, they perform different functions. Because the cost functions for MMA operate in separate dimensions, MMA can accomplish complete equalization, while CMA can only do it partially.

3.2.2 Nonsquare Constellations

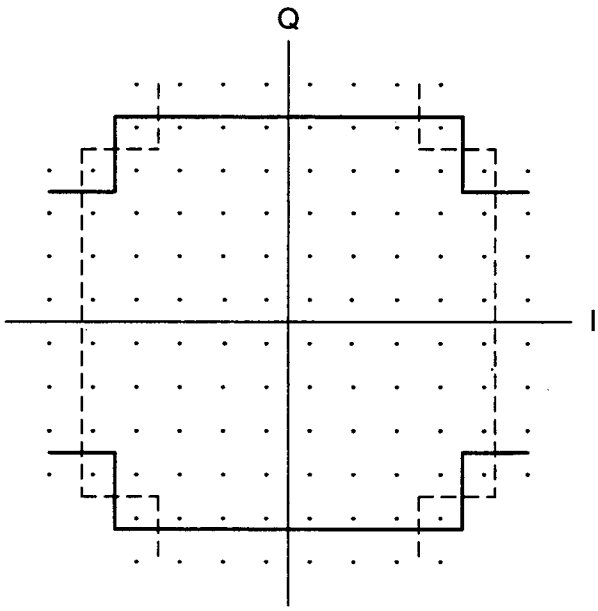
The use of one constant R is the simplest application of MMA. The use of multiple moduli for nonsquare constellations is considered as the first extension of MMA. In this case, multiple moduli are used when a nonuniform symbol distribution occurs across each dimension, as shown in Fig. 3.6. For nonsquare constellations, R is not a constant, instead, multiple moduli R are required because the output samples y_n and $\tilde{y}_n$ have different distributions depending on their values. In the case of nonsquare constellations, the signal constellation is designed in such a way that the discrete levels for the symbols a_n and b_n do not have the same probability of occurence. For instance, for 128-CAP we have a number of symbol levels $m = 6$. However, these levels are divided into two subsets, which are $a_{n,1} = \{\pm1, \pm3, \pm5, \pm7\}$, and $a_{n,2} = \{\pm7, \pm9\}$. The two subsets of symbols have a different probability of occurence, as shown in Fig. 3.6. If one constant R is used for all the symbols, the knowledge of the symbol statistics is not properly used, and the equalizer may converge to wrong solutions. One wrong solution is called the 144-point solution [29], which occurs when the in-phase and quadrature filters converge to impulse responses which are offset by a symbol period.



a. Moduli for In-Phase Dimension



b. Moduli for Quadrature Phase Dimension



c. Moduli for Both Dimensions

Figure 3.6: MMA moduli for 128-point signal constellation

The MMA cost function for nonsquare constellations minimizes the dispersion of the equalizer's output samples y_n and $\tilde{y}_n$ with respect to piecewise straight lines. Because there are two subsets of symbols, we need two constants R to provide efficient statistical description of the symbols a_n. As shown in Fig. 3.6, the two constants R are represented by two straight lines. The cost functions are given by the piecewise nonlinear functions

$$CF_I = E[(y_n^L - R_1^L)^2] \;\; if \;\; |\tilde{y}_n| < K \qquad CF_I = E[(\tilde{y}_n^L - R_2^L)^2] \;\; if \;\; |\tilde{y}_n| > K \tag{3.32}$$

$$CF_Q = E[(y_n^L - R_1^L)^2] \;\; if \;\; |y_n| < K \qquad CF_Q = E[(\tilde{y}_n^L - R_2^L)^2] \;\; if \;\; |y_n| > K \tag{3.33}$$

$$\tag{3.34}$$

where K refers to the boundary separating the two sets of symbols. The use of multiple moduli in MMA for nonsquare constellations is illustrated in Fig. 3.6. This figure shows a 128-CAP signal constellation, for which blocks of seven bits are mapped into $2^7 = 128$ complex symbols. By using the 12 symbol levels $\{\pm1, \pm3, \pm5, \pm7, \pm9, \pm11\}$ we can define $12 \times 12 = 144$ complex symbols, so that 144-128=16 points are redundant and can be eliminated. For typical applications where 128-CAP is used, the four outer points in each quadrant are eliminated. This reduces the peak-power-to-average-power ratio of the CAP signal, which is always a desirable feature in practice. For the 128-CAP constellation shown in Fig. 3.6, a probability of two-thirds is used for symbol set one, $|y_n| < K$. Then K is calculated as

$$K = \frac{2}{3} * 2m \tag{3.35}$$

where m indicates the number of symbol levels of C-CAP

$$m = \frac{3}{4}\sqrt{\frac{C}{2}} \tag{3.36}$$

and C refers to the number of constellation points. $K = 8$ is obtained for 128-CAP. It means that symbol set one has six symbol levels for $y_n < 8$, and symbol set two has four symbol levels for $y_n > 8$.

In the following, the calculation of the moments of the symbols a_n is illustrated for 128-CAP. The computation of the moments of symbols is considered in one quadrant for the in-phase dimension.

Let us define symbol set one for $y_n < K$ and symbol set two for the rest. The probability P_1 for symbol set one is computed as

$$P_1 = \frac{4}{C} * m = \frac{3}{16} \tag{3.37}$$

and P_2 for each symbol in set two is computed as

$$P_2 = \frac{4}{C} * \frac{2m}{3} * m = \frac{1}{8} \tag{3.38}$$

The moduli R_1 and R_2 for nonsquare constellation are computed from equation 3.26 by evaluating the moments of the symbols over the set of symbol levels to which a given modulus applies (illustrated below). As an example, consider Fig. 3.6, which shows the moduli for the in-phase dimension and which applies to the real symbols a_n of a 128-CAP signal constellation. The moments of the symbols can be computed by considering the first quadrant only. Consider the subset of 24 symbols in this quadrant which applies to R_1. For these symbols $a_n = 1, 3, , 5, 7, 9, 11$ and $b_n = 1, 3, 5, 7$. So that each value of a_n occurs with probability $4/24 = 1/6$. Similarly, the R_2 subset has 8 symbols for which $a_n = 1, 3, 5, 7$ and $b_n = 9, 11$, so that each value of a_n occurs with probability $2/8 = 1/4$. Thus, the variance of the symbols becomes

$$E[a_n^2] = \frac{1}{6}(1^2 + 3^2 + 5^2 + 7^2 + 9^2 + 11^2) \approx 47.67 \quad \text{for } R_1 \tag{3.39}$$

$$E[a_n^2] = \frac{1}{4}(1^2 + 3^2 + 5^2 + 7^2) = 21 \quad \text{for } R_2 \tag{3.40}$$

Other moments for the symbols are computed in a similar fashion. The two moduli are given by $R_1 = 9.2$ and $R_2 = 6.1$ for 128-CAP. The probability of converging to a wrong 144-point solution is significantly reduced by using two moduli for nonsquare constellations.

Because of the use of multiple moduli, the constants R can provide adequate knowledge of the statistics of the symbol distribution. The number of constants R is equal to the number of symbol sets having different statistics.

3.2.3 ISI Optimization

In this section it is shown that minimization of the MMA cost function removes ISI at the output of the equalizer.

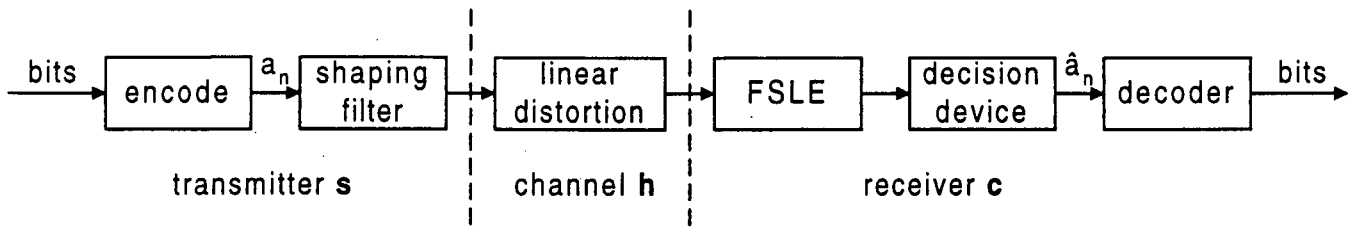


Figure 3.7: Communication link

where $\mathbf{a}_n$ is the vector of in-phase symbols and the vector $\mathbf{w}$ denotes the vector of symbol-spaced samples of the overall impulse response of the transmission link

$$\mathbf{w} = \mathbf{s}_n \otimes \mathbf{h}_n \otimes \mathbf{c}_n \tag{3.42}$$

The input and channel vectors $\mathbf{a}_n$ and $\mathbf{w}$ are defined as follows

$$\mathbf{a}_n^T = [a_{n+k}, \ldots, a_n, \ldots, a_{n+k-N}] \tag{3.43}$$

$$\mathbf{w}^T = [w_0, w_1, \ldots, w_N] \tag{3.44}$$

where $N+1$ is the number of samples. With a perfectly equalized channel and no ISI at the input of the slicer, the channel vector $\mathbf{w}$ has only one nonzero entry equal to one and can be written as:

$$\mathbf{w}_0^T = [\underbrace{0, \ldots, 0}_{k}, 1, 0, \ldots, 0] \tag{3.45}$$

where k indicates the channel and equalizer delay expressed in symbol periods. It is easily verified from (3.43) and (3.45) that

$$y_n = \mathbf{w}_0^T \mathbf{a}_n^T = a_n \tag{3.46}$$

So that the equalizer's output is indeed ISI free. Using (3.41), the second order expectation of the sample y_n is

$$E[y_n^2] = E[\mathbf{w}^T \mathbf{a}_n \mathbf{w}^T \mathbf{a}_n] = E[\mathbf{w}^T \mathbf{a}_n \mathbf{a}_n T \mathbf{w}] = \mathbf{w}^T E[\mathbf{a}_n \mathbf{a}_n^T] \mathbf{w} \tag{3.47}$$

With the usual assumption that different symbols are uncorrelated we get

$$E[\mathbf{a}_n \mathbf{a}_n^T] = E[a_n^2] * \mathbf{I} \tag{3.48}$$

where $\mathbf{I}$ is the unity diagonal matrix, so that

$$E[y_n^2] = E[a_n^2] \mathbf{w}^T \mathbf{w} = E[a_n^2] \sum_{k=0}^{N} w_k^2 \tag{3.49}$$

In order to show that minimization of the cost function results in zero ISI, we can show that

$$E[(y_n^2 - R^2)^2] \geq E[(a_n^2 - R^2)^2] \tag{3.50}$$

and that the minimum is achieved when the channel vector $\mathbf{w}$ satisfies (3.45). We now provide two such proofs, one for a constrained problem and one for a nonconstrained problem. A result which will be useful in the following analysis is:

$$(\sum_i w_i^2)^2 \geq \sum_i w_i^4 \quad \rightarrow \quad (\sum_i w_i^2)^2 = \sum_i w_i^4 + \delta \qquad \delta \geq 0 \tag{3.51}$$

Equality in (3.51) holds if and only if one of the terms w_i is nonzero and all the other terms are zero. (Equality also holds for the trivial case where all the terms w_i are zero.)

Constrained problem

In this scenario, we assume that the channel vector $\mathbf{w}$ can be written as

$$\mathbf{w}^T = [x, \ldots, x, 1, x, \ldots, x] \tag{3.52}$$

That is, one of the entries of the vector is constrained to remain equal to one, and the other entries can take any value. This assumption is a good approximation of what is done in practice when

the gain of the AGC and the initial values of the equalizer are properly chosen. It is easily verified that, with the constraint in (3.52) the following holds:

$$\sum_{k=0}^{N} w_k^2 \geq 1 \tag{3.53}$$

and the equal sign holds if and only if the channel vector satisfies the ISI free condition in (3.45).

The condition in (3.50) can be written as

$$E[y_n]^4 - 2E[y_n^2]R^2 + R^4 \geq E[a_n^4] - 2E[a_n^2]R^2 + R^4 \tag{3.54}$$

$$E[y_n^4] - 2E[y_n^2]R^2 \geq E[a_n^4] - 2E[a_n^2]R^2 \tag{3.55}$$

Replacing R by its value and using (3.49) in (3.55), we can further simplify the equation to

$$E[y_n^4] \geq E[a_n^4](2\sum_k w_k^2 - 1) \tag{3.56}$$

From the inequality in (3.53), we then have

$$E[y_n^4] = E[a_n^4][2\sum_k w_k^2 - 1] \geq E[a_n^4] \tag{3.57}$$

and equality can only hold if the channel vector satisfies the ISI free condition in (3.45).

Unconstrained problem

We now remove the constraint in (3.52) of keeping one entry of the channel vector equal to one. Using results given in [25], it is possible to show that

$$K(y_n) = E[y_n^4] - 3E^2[y_n^2] \tag{3.58}$$

$$K(a_n) = K(a_n)\sum_k w_k^4 \tag{3.59}$$

where $K(y_n)$ and $K(a_n)$ are called the kurtosis of the equalizer output samples and symbols, respectively. We can use (3.49) and (3.59) to solve for $E[y^4]$ in (3.59) and we get:

$$E[y_n^4] = E[a_n^4]\sum_k w_k^4 - 3E^2[a_n^2][\sum_k w_k^4 - (\sum_k w_k^2)^2] \tag{3.60}$$

Using (3.51) on the right in (3.60) we get:

$$E[y_n^4] = E[a_n^4]\sum_k w_k^4 + 3\delta E^2[a_n^2] \tag{3.61}$$

Replacing R by its value and using (3.59), the 1-D cost function of MMA in (3.23) can be written as

$$E[(y_n^2 - R^2)^2] = E[y_n^4] - 2E[a_n^4]\sum_k w_k^2 + R^4 \tag{3.62}$$

Using (3.61) in (3.62), we get

$$\begin{aligned} E[(y_n^2 - R^2)^2] &= E[a_n^4]\sum_k w_k^4 + 3\delta E^2[a_n^2] - 2E[a_n^4]\sum_k w_k^2 && (3.63)\\ &= E[a_n^4][\sum_k w_k^4 - 2\sum_k w_k^2] + 3\delta E^2[a_n^2] + R^4 && (3.64) \end{aligned}$$

Replacing $\sum_k w_k^4$ in the above equation by its value on the right in (3.51), we have

$$\begin{aligned} E[(y_n^2 - R^2)^2] &= E[a_n^4][(\sum_k w_k^2)^2 - 2\sum_k w_k^2 - \delta] + 3\delta E^2[a_n^2] + R^4 && (3.65)\\ &= E[a_n^4][(\sum_k w_k^2 - 1) - 1 - \delta] + 3\delta E^2[a_n^2] + R^4 && (3.66)\\ &= E[a_n^4][(\sum_k w_k^2 - 1) - 1] - \delta[E[a_n^4] - 3E^2[a_n^2]] + R^4 && (3.67) \end{aligned}$$

From (3.67) we conclude that the following holds:

$$E[(y_n^2 - R^2)^2] \geq -E[a_n^4] + R^4 = E[(a_n^2 - R^2)^2] \tag{3.68}$$

if

$$(\sum_k w_k^2 - 1)^2 \geq 0 \quad \textit{and} \quad E[a_n^4] - 3E^2[a_n^2] \leq 0 \tag{3.69}$$

The left condition is always true. The right condition says that the kurtosis of the symbols has to be always negative. Equality can only hold if $\sum_k w_k^2 = 1$ and $\delta = 0$ in (3.67). From (3.51), this implies that only one w_k can be nonzero and $\sum_k w_w^2 = 1$ implies that the nonzero term has to be one, so that the channel vector $\mathbf{w}$ has to have the ISI-free form given in (3.45)

To show that the kurtosis of the symbols used in typical signal constellation is always negative we can use the results obtained in Appendix B. The second and fourth-order moments of the symbols are obtained from (0.23), (0.25) and (0.26), and we get

$$\begin{aligned} K(a_n) &= E[a_n^4] - 3E^2[a_n^2] = \frac{1}{15}(4m^2 - 1)(12m^2 - 7) - 3 * \frac{1}{9}(4m^2 - 1)^2 && (3.70)\\ &= \frac{1}{15}(4m^2 - 1)[12m^2 - 7 - 5(4m^2 - 1)] && (3.71)\\ &= -\frac{2}{15}(4m^2 - 1)(4m^2 + 1) && (3.72) \end{aligned}$$

It is obvious that the values in the brackets are always larger than zero for $m \geq 1$. As a result

$$K(a_n) < 0 \quad \text{for} \quad m \geq 1 \tag{3.73}$$

and this completes the proof that minimization of the MMA cost function leads to ISI free output samples of the equalizer.

3.3 Comparison of the Three Algorithms

So far, three different blind equalization algorithms have been studied. The preference for each of them depends on the application. In this section, the algorithms are analyzed and compared in terms of convergence, reliability and complexity.

Algorithmic structures

First of all, we study the relation between the three algorithms. For convenience, the cost functions of the three algorithms are repeated as follows.

$$MSE = E[|Y_n - A_n|^2] \tag{3.74}$$

$$CF_{rca} = E[|Y_n - Rsgn(Y_n)|^2] \tag{3.75}$$

$$CF_{cma} = E[(|Y_n|^2 - R^2)^2] = E[(y_n^2 + \tilde{y}_n^2 - R^2)^2] \tag{3.76}$$

$$CF_{mma} = E[(y_n^2 - R^2)^2 + (\tilde{y}_n^2 - R^2)^2] \tag{3.77}$$

where we have also included the MSE, which is the cost function for the LMS algorithm. Note that the MSE and the cost function for RCA use second-order statistics of the samples Y_n and the other two use fourth-order statistics.

The tap updating algorithms are obtained by taking the gradient of the cost functions with respect to the tap coefficients, and then using the nonaveraged gradients in a stochastic gradient algorithm. For the phase-splitting equalizer, all the tap updating algorithms for the in-phase coefficients, for example, have the following generic form:

$$\mathbf{c}_{n+1} = \mathbf{c}_n - \mu k_n \mathbf{r}_n \tag{3.78}$$

where $\mathbf{c}_n$ is the in-phase tap vector, $\mathbf{r}_n$ is the vector of A/D samples, and k_n is a correction term, which is correlated with the vector $\mathbf{r}_n$ of A/D samples and is different for each algorithm.

Specifically, we have

$$LMS \quad : \quad k_n = y_n - a_n \tag{3.79}$$

$$RCA \quad : \quad k_n = y_n - Rsgn(y_n) \tag{3.80}$$

$$CMA \quad : \quad k_n = (y_n^2 + \tilde{y}_n^2 - R^2)y_n \tag{3.81}$$

$$MMA \quad : \quad k_n = (y_n^2 - R^2)y_n \tag{3.82}$$

The cost function for RCA and the LMS algorithm are the same, except that the sliced outputs $\hat{a}_n = Rsgn(y_n)$ are obtained with a four-point slicer. Otherwise, the implementation of the two algorithms is identical. This makes RCA easy to implement and easy to analyze, and explains why RCA is widely used in practical applications.

It is interesting to note that CMA and MMA are identical for one-dimensional transmission schemes. For example, assume that we only have the in-phase dimension so that $\tilde{y}_n$ in (3.76) and (3.77) is zero. We then have for both cases

$$CF_I = E[(y_n^2 - R^2)^2] \tag{3.83}$$

We see that the cost functions of CMA and MMA are identical under certain conditions. The two algorithms have the same statistical orders and the same algorithmic structures. Except for linear channel phase equalization, the two algorithms can achieve the same performance.

The major difference between CMA and MMA is the way they use the complex samples $Y_n = y_n + j\tilde{y}_n$. CMA treats the samples y_n and $\tilde{y}_n$ as one complex sample Y_n. This leads to the following result

$$CF_{cma}(\theta) = E[|Y_n e^{j\theta}|^2 - R^2] = E[|Y_n|^2 - R^2] \tag{3.84}$$

Note that the cost function CF_{cma} is independent of θ. The cost function of MMA is given by

$$CF_{mma}(\theta) = E[(y_n^2 \cos^2\theta - R^2)^2 + (\tilde{y}_n^2 \sin^2\theta - R^2)^2] \tag{3.85}$$

Note that in the above equation, the cost function CF_{mma} is a function of θ. CMA is able to converge to any rotated version of the signal constellations. In contrast, for MMA, only one θ minimizes the cost function. Therefore, with MMA, ISI equalization and phase recovery can be done jointly, and the additional rotator is not required to tune the constellation to its final position.

In summary, even though CMA and MMA are somewhat similar, they produce completely different results. Moreover, they associate with different problems. The use of CMA is expensive because a rotator is required after the equalizer in steady-state operation. However, CMA is attractive because of its ability to converge without creating diagonal solutions. Particularly, in the case of the phase-splitting equalizer structure, the two filters converge independently without any coupling between the two channels. In this case, the possibility of creating diagonal solutions exists for any algorithm that simultaneously equalizes magnitude, phase, and phase offset as well.

The calculation of R can be found in Appendix B. The values of the constants R for the three algorithms are listed in Table 3.1 for square and nonsquare CAP applications.

Algorithm/CAP	4-CAP	16-CAP	32-CAP	64-CAP	128-CAP	256-CAP	512-CAP	1024-CAP
RCA	1	2.50	3.64	5.25	7.45	10.63	15.00	21.31
MMA	1	2.86	4.32	6.08	8.88	12.34	17.87	24.76
CMA	1.414	3.63	5.11	7.62	10.49	15.39	21.11	30.9
MMA R_1	–	–	4.49	–	9.22	–	18.55	–
MMA R_2	–	–	2.86	–	6.08	–	12.34	–

Table 3.1: Constant R for symbol levels $\pm 1, \pm 3, \ldots, \pm 2m - 1$

Note that $R_{rca} < R_{mma} < R_{cma}$. It shows that the constant R is larger for the second-order algorithm RCA than those of the fourth-order algorithms of MMA and CMA, and R is larger for the two-dimensional algorithm CMA than that of one-dimensional one MMA.

3.4 Summary of Results

The investigation of the three algorithms shows that each algorithm has its advantages and disadvantages. The major attributes of each algorithm can be summarized as follows.

The major attraction of RCA is its simplicity and ease of implementation. Its implementation is the same as that of the LMS algorithm. However, good performance is difficult to obtain because

of the limitations that result from the usage of second-order statistics only. The reliability is poor and the probability of creating diagonal solutions is the highest among the three algorithms.

The advantages of CMA are its robustness and reliability, particularly to diagonal solutions. The main disadvantage of CMA is its complexity. With CMA, equalization can only be partially done, and a rotator is needed to complete the whole equalization, which significantly increases the cost of implementation for steady-state operation. It should be pointed out, however, that there are some applications, such as voiceband and cable modems, where rotation of the constellation is required anyway for other purposes, such as tracking frequency offset introduced in the channel. In this case, the need to do rotation does not increase the cost of implementation, and CMA becomes a very attractive approach. MMA is a new blind equalization algorithm that was first proposed in [29] [30]. It is a two-dimensional algorithm that can be considered to be a superposition of two independent one-dimensional algorithms similar to CMA. It uses high-order statistics so that it has a fast convergence rate, and it is more immune to diagonal solutions than RCA. However, when magnitude and phase offset equalization are done at once, the possibility of occasionally converging to a diagonal solution still exits. Solutions to help in these situations are discussed in related work [30].

One of the major advantages of MMA over both RCA and CMA is its flexibility. Because MMA can easily use multiple moduli, which is difficult to do with RCA and CMA, it can be adapted to applications for which RCA and CMA are not very effective. One example, presented in this chapter, is the case of nonsquare constellation. Another example, discussed in the next chapter, is that of very dense signal constellation.

The characteristics of the three algorithms are summarized in Table. 3.2. We see that none of the algorithms has all the desirable features. By using MMA, we can obtain adequate reliable and fast convergence at relatively low cost.

Algorithm	Reliability	Complexity	Convergence rate*	Flexibility
RCA	low	low	second fastest	low
MMA	high	medium	fastest	very high
CMA	very high	high	slowest	low

* For the ATM LAN application.

Table 3.2: Characteristics of the blind equalization algorithms

Table 3.2: Characteristics of the blind equalization algorithms

Chapter 4

Generalized MMA (GMMA)

One aspect of the flexibility provided by MMA was demonstrated in the previous chapter, where it was shown how it can easily be modified to accommodate nonsquare signal constellations and take advantage of the fact that these constellations have subsets of symbol levels with different statistics. These modifications facilitate the eye opening of nonsquare constellations and greatly decrease the probability of converging to wrong solutions, especially the 144-point solution for 128-CAP. Both RCA and CMA cannot easily be modified to accommodate nonsquare constellation and get similar benefits.

Another aspect of the flexibility provided by MMA is demonstrated in this chapter, where we show how it can be modified to ease the eye opening of signal constellations with a very large number of symbol levels. This is achieved by dividing the complex plane of in-phase and quadrature output samples of the equalizer into several disjoint regions, which all have their own cost functions and moduli. This modified MMA algorithm is called generalized MMA (GMMA).

GMMA can open the eye of very large signal constellations for which RCA, CMA and standard MMA would not be effective. This is demonstrated by the analysis given in Appendix C, where the relationship between the MSE and the minimum value of the standard MMA cost function is discussed. GMMA can handle very large signal constellations because the residual values of the cost functions obtained with GMMA can be made much smaller than the residual values obtained with the other algorithms. The definition of the various subsets of equalizer output samples and

corresponding moduli used by GMMA is not straightforward, and has to be made carefully if one wants to get the benefits provided by this algorithm. To address this problem, we present here an iterative algorithm, which uses an equal energy type of principle and leads to the definition of near-optimum subsets of equalizer output samples and corresponding moduli.

4.1 Cost Function Analysis

The study of cost functions is essential to the understanding of the convergence of a blind equalizer. Cost functions for blind algorithms behave differently from that of the standard LMS algorithm. For the standard LMS algorithm, the cost function minimizes the error e_n between the equalizer's output signals Y_n and an unknown sequence of transmitted symbols A_n

$$CF = E[(Y_n - A_n)^2] = E[(y_n - a_n)^2 + [(\tilde{y}_n - b_n)^2] = E[e_n^2(LMS) + \tilde{e}_{r,n}^2(LMS)] \tag{4.1}$$

where $Y_n = y_n + j\tilde{y}_n$, and $A_n = a_n + jb_n$. When a training sequence is available, the errors $e_{r,n}(LMS)$ and $\tilde{e}_{r,n}(LMS)$ can be calculated in a meaningful way from the equalizer's outputs. In this case, we say that channel equalization is done with ideal reference. However, without ideal reference, the unknown inputs a_n and b_n have to be decided at the slicers [30]. Initially, with this approach, a blind equalizer makes too many wrong decisions because the signals are severely corrupted by intersymbol interference (ISI). For severe ISI channels, a blind equalizer cannot converge if the LMS algorithm is used at the beginning of start-up.

In order to reduce the probability of making wrong decisions, several blind algorithms are proposed to make an equalizer converge during blind start-up [30]. In the case of MMA, the cost function minimizes the dispersion of the constellation around linear contours, and it is given by

$$CF = E[(y_n^2 - R^2)^2 + (\tilde{y}_n^2 - R^2)^2] = E[e_{r,n}^2(CF) + \tilde{e}_{r,n}^2(CF)] \tag{4.2}$$

where the constant R is given by

$$R^2 = \frac{E[a_n^4]}{E[a_n^2]} \tag{4.3}$$

Comparing the two cost functions in (4.1) and (4.2), we can see that errors have different interpretations for the LMS algorithm and MMA. The difference can also be visualized in Fig. 4.1 where

the errors $e_{r,n}(LMS)$ and $e_{r,n}(CF)$ are shown. In the LMS algorithm, the error for the in-phase filter $e_{r,n}(LMS)$ is defined as

$$e_{r,n}(LMS) = y_n - a_n \tag{4.4}$$

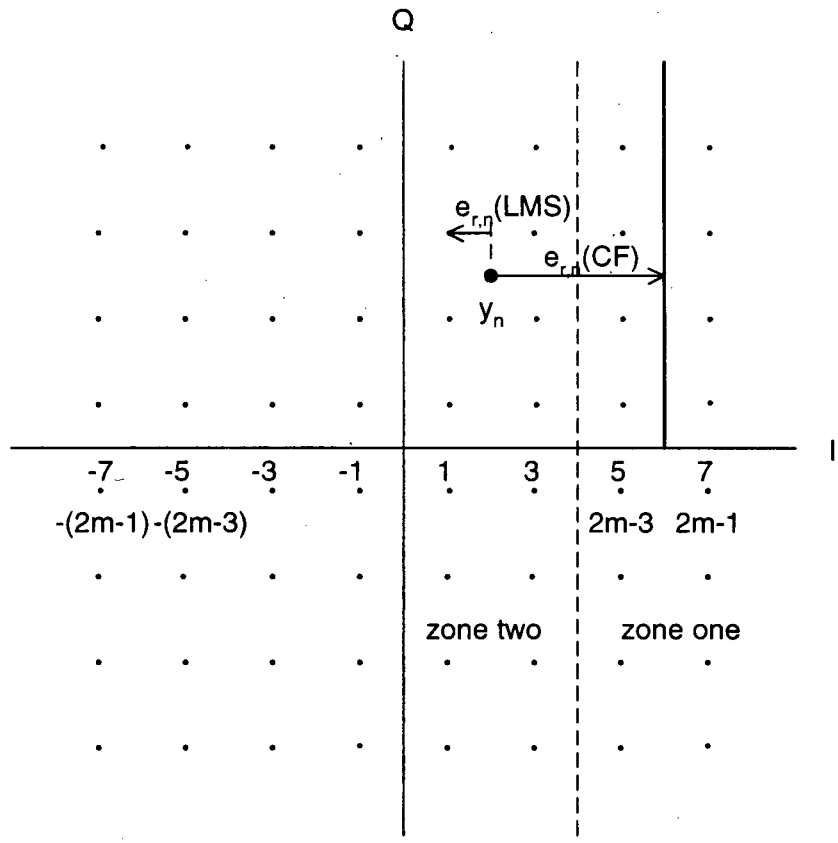


Figure 4.1: Errors for CF and LMS algorithm

The taps are updated in the opposite direction of the gradient

$$\mathbf{c}_{n+1} = \mathbf{c}_n - \mu e_{r,n}(LMS)\mathbf{r}_n = \mathbf{c}_n - \mu(y_n - a_n)\mathbf{r}_n \tag{4.5}$$

When y_n and a_n represent the input and output of the slicer, the error $e_{r,n}(CF)$ used during tap updating is equivalent to the error measured at the slicer. The error $e_{r,n}(CF)$ is defined differently. Note that the LMS algorithm uses second-order statistics of the signals, whereas MMA uses fourth-order statistics. In order to compare the two algorithms, we use the MMA with $L = 1$.

For one-dimensional MMA with $L = 1$, the error $e_{r,n}(CF)$ of the generalized MMA becomes

$$e_{r,n}(CF) = |y_n| - R \quad . \tag{4.6}$$

The filter's taps are updated as follows

$$\mathbf{c}_{n+1} = \mathbf{c}_n - \mu e_{r,n}(CF)\mathbf{r}_n = \mathbf{c}_n - \mu(|y_n| - R)\mathbf{r}_n \tag{4.7}$$

The taps are not exactly updated in the direction of the slicer's errors. Instead, error minimization is done with reference to the constant R that has statistical information about the real symbols a_n. The direction of filter adaptation depends on the constant R, which depends on m. In summary, the error $e_{r,n}(LMS)$ is a well-defined quantity. An equalizer can converge to optimal solutions when errors are directly calculated with respect to the inputs and outputs of the slicer. In contrast, the error $e_{r,n}(CF)$ has only a statistical meaning. An equalizer is not always guaranteed to converge to optimal solutions in terms of mean-square error (MSE).

For blind algorithms, we usually have $CF \neq 0$ when an equalizer has converged. That is, residual values of the cost function exist. At this stage, in order to understand the MMA blind algorithm, it is necessary to investigate what are the residual values of the cost function CF. For a blind start-up with a perfect convergence, $y_n \rightarrow a_n$. Consequently as was shown in the previous chapter, the cost function CF converges to CF_{an}

$$CF = E[(y_n^2 - R^2)^2] \quad \rightarrow \quad CF_{an} = E[(a_n^2 - R^2)^2] \tag{4.8}$$

The cost function CF_{an} is then expanded, factorized and simplified as follows

$$\begin{aligned}
CF_{a_n} &= E[(a_n^2 - R^2)^2] \\
&= E[a_n^4 - 2a_n^2R^2 + R^4] \\
&= E[a_n^4 - 2a_n^2\frac{E[a_n^4]}{E[a_n^2]} + R^4] \\
&= E[a_n^4] - 2E[a_n^2]\frac{E[a_n^4]}{E[a_n^2]} + R^4 \\
&= R^4 - E[a_n^4] \\
&= \frac{E^2[a_n^4]}{E^2[a_n^2]} - E[a_n^4]
\end{aligned}$$

Note that only the analysis for the in-phase dimension is provided, and that the same analysis applies to the quadrature phase dimension. In order to evaluate the cost function, we need to express CF_{an} as a function of the number of symbol levels m. Calculations of moments of data symbols can be found in Appendix B. For MMA, the constant R is given by

$$R^2 = \frac{12m^2 - 7}{5} \tag{4.9}$$

and the fourth moment of the symbols a_n is given by

$$E[a_n^4] = \frac{1}{15}(4m^2 - 1)(12m^2 - 7) \tag{4.10}$$

Then the cost function CF_{an} can be rewritten as

$$\begin{aligned} CF_{an} &= R^4 - E[a_n^4] && (4.11)\\ &= (\frac{12m^2 - 7}{5})^2 - \frac{1}{15}(4m^2 - 1)(12m^2 - 7) && (4.12)\\ &= \frac{16}{75}(12m^2 - 7)(m^2 - 1) && (4.13)\\ &= \frac{16}{75}(12m^4 - 19m^2 + 7) && (4.14) \end{aligned}$$

Eq. 4.14 gives a simple way to express the cost function after convergence.

Steady-state values of the cost function CF_{an} can then be easily calculated. The number of symbol levels m (in magnitude) can be computed from the number of constellation points C for C-CAP

$$m = \frac{\sqrt{C}}{2} \tag{4.15}$$

Eq. 4.14 shows that $CF_{an} = 0$ for $m = 1$, and $CF_{an} \neq 0$ for $m \neq 1$. For instance, calculating CF_{an} gives the following results: $CF_{an} = 0$ for 4-CAP with $m = 1$, $CF_{an} = 14.2$ dB for 16-CAP with $m = 2$, and $CF_{an} = 27.7$ dB (we choose dB in measurement) for 64-CAP with $m = 4, \ldots$. It means that the optimum convergence for a blind equalizer can only be achieved for 4-CAP with $m = 1$. Residual values of the cost function CF_{an} significantly increase with increasing m. Ultimately, for large values of the number m, residual values of CF_{an} become so large that a blind equalizer fails to converge.

Residual values of CF_{an} are increasing functions of the number m, and an equalizer's convergence is directly affected by those values. In conclusion, the reliability of a blind algorithm is highly

degraded with increasing m. When the residual values of CF_{an} increase beyond some quantity, the eye diagram fails to open. It has been found that a standard MMA is only effective for CAP applications with m less that eight which corresponds to 256-CAP.

4.2 Principle of Multimodulus

With MMA, the use of more than one modulus for each dimension was originally proposed for nonsquare constellations [29]. MMA minimizes the dispersion of the equalizer's output samples y_n around piecewise linear contours. The general form of the in-phase MMA cost function for $L = 2$ is given by

$$CF = E[(y_n^2 - R^2(y_n))^2] \tag{4.16}$$

where values of $R(y_n)$ are determined by the distribution function of the samples y_n. For square constellations, R becomes a single constant. For nonsquare constellations, however, $R(y_n)$ takes multiple values that depend on the number of sets of symbols a_n with different statistics. For instance, for 128-CAP, the symbols are transmitted in two sets, whose values are given by $a_1 = \{\pm 1, \pm 3, \pm 5, \pm 7\}$ and $a_2 = \{\pm 9, \pm 11\}$. Two constants R_1 and R_2 are necessary because the sets of symbols $a_{1,n}$ and $a_{2,n}$ have different probability distributions. The study in [29] shows that the use of multiple moduli leads to a reduction of the number of wrong solutions. Particularly, the probability of creating the 144-point solution for 128-CAP is significantly reduced when two moduli R are used.

Further study has shown that, even for square constellations, the use of multiple moduli can improve the reliability of convergence. The ultimate knowledge one can have about the statistics of the transmitted symbols is to know exactly which symbols have been transmitted. This can be the case at start-up only when a training sequence is utilized, or in steady-state, when the eye is open and wrong decisions occur rarely. In this case, the MSE is the best cost function and the corresponding LMS algorithm is the best tap updating algorithm. However, the LMS algorithm cannot be used during blind start-up because of excessive ISI which creates too many wrong decisions. At the low end of our knowledge of the statistics of the transmitted symbols are single one or two-dimensional contours around which the dispersion of the symbols is minimized.

Usage of this type of knowledge leads to MMA and CMA. The advantages of using this type of approach is that most output samples of the equalizer are meaningful. That is, there are many less wrong decisions.

The difficulty with using single contours is that the correction terms used in the tap updating algorithms become very large compared to the spacing between adjacent symbols. As a result, it becomes increasingly difficult to open the eye when the number m of symbol levels increases. For very large values of m, opening of the eye would require such a small step size that the convergence time would be unacceptable. This issue is discussed in more detail in Appendix C.

Minimizing the dispersion of subsets of equalizer output samples y_n around several contours minimizes the maximum values that can be assumed by the correction terms in the tap updating algorithms. This facilitates the opening of the eye. However, with such an approach, more wrong correction terms are going to be used during initial blind start-up. That is, because of ISI, some samples y_n will appear in a wrong area in the complex plane and will be used to compute a correction term with respect to the wrong modulus.

In many ways, this problem is similar to the problem experienced by the LMS algorithm during initial blind start-up. The difference is that LMS would almost always use the wrong correction term when there is severe ISI, whereas GMMA, if it is well designed, will use a large enough correct correction terms to allow the eye to open.

Fig. 4.1 shows the two errors used during filter adaptation. Let ∇e represent the difference between $e_{r,n}(LMS)$ and $e_{r,n}(CF)$, that is, $\nabla e = e_{r,n}(CF) - e_{r,n}(LMS)$. The constant R statistically describes a_n in two different ways, and it depends on the distribution of y_n. We divide the samples y_n into two zones. For instance, in 64-CAP, we compute that $R = 6$. As shown in Fig. 4.1, we divide data into two zones above and below the dotted line corresponding to $y_n = 4$. Let δ denotes a small number. When the samples y_n are distributed in zone one, $\nabla e =\leq \delta$. Correspondingly, the residual value of the cost function CF_{an} is smaller. This makes the equalizer converge almost to optimal solution. When the samples y_n are located in zone two, $\nabla e > \delta$. The large ∇e gives the large residual value of the cost function CF_{an}. Consequently, it makes the opening of the eye difficult.

We conclude that during blind start-up, ∇e contributes different effects to steady-state convergence depending on the distribution of a_n. When samples a_n are distributed near the constant R, the absolute values of the two errors are equal to or close to zero, and the equalizer can approximately converge to the optimal values that can be achieved with the LMS algorithm. Otherwise, the equalizer converges to a solution with residual MSE. At this point, multiple moduli are required so that the use of the constant R can represent knowledge of statistics of symbols a_n efficiently under the condition of minimal wrong decisions. We conclude that multiple moduli are also necessary for square constellations. In fact, the use of generalized MMA is just a natural extension of MMA. In MMA for nonsquare constellation, multiple moduli are required to satisfy nonuniform distribution of y_n, whereas the algorithm of multilevel moduli is designed for uniform distribution of y_n to improve convergence reliability.

We now discuss the concept of sample subsets used in GMMA. A sample subset is defined as

$$y_n = \{y_{n,i}\} \quad i = 1, \ldots, I \tag{4.17}$$

Where $I < m$. The cost function minimization is done with respect to each sample subset $y_{n,i}$

$$CF = \frac{1}{I}\sum_{i=1}^{I} CF_i = \frac{1}{I}\sum_{i=1}^{I} E[(y_{n,i}^2 - R_i^2)^2] \tag{4.18}$$

The cost function is reduced to the cost functions CF_i corresponding to the subset $y_{n,i}$. When an equalizer converges, we have $y_n \to a_n$, or, in terms of sample subset, we have $\{y_{n,i}\} \to \{a_{n,i}\}$. The minimum of the cost function in (4.18) is given by

$$CF_{an} = \frac{1}{I}\sum_{i=1}^{I} CF_{an,i} = \frac{1}{I}\sum_{i=1}^{I} E[(a_{n,i}^2 - R_i^2)^2] \tag{4.19}$$

when the outputs of the equalizer have converged to constellation points $a_{n,i}$. In this case, the constant R_i refers to the set of symbols $a_{n,i}$.

When using sample subsets, the overall system performance is determined by the individual performance for each data subset, so that a performance criterion is required for the system. The analysis in the previous section shows that the reliability of convergence depends on the residual values of the cost functions in steady state. It is natural then to use the equal cost function minimization procedure. That is, the cost functions $CF_{an,i}$ are minimized in such a way that the

residual value of each $CF_{an,i}$ is reduced to the same amount. With equal cost function minimization, we have

$$CF_{an,i} = C \tag{4.20}$$

where C is a suitably chosen constant. With GMMA, the equalizer tries to reduce the residual value of the cost function in order to achieve a good eye opening.

4.3 Parameter Design

In this section, we show how to select the subsets of samples used for GMMA. With MMA, only one constant R needs to be determined for a given constellation. Unlike MMA, several parameters are needed to describe sample subsets in GMMA. In addition, because the sample subsets $y_{n,i}$ are undefined at the beginning of the algorithm, a numerical solution cannot be easily derived. Our study shows that cost functions can be expressed as a function of the number m [30]. We can also express the cost function $CF_{an,i}$ as a function of m_i, where $2m_i - 1$ is the highest symbol level for sample subset i, as shown in Fig. 4.2. With one unknown m_i, the cost function $CF_{an,i} = C$ can be numerically solved. Other parameters can be solved as well if they can be expressed as a function of m_i.

In summary, m_i is the fundamental parameter in the overall system design because it can be directly derived for a given cost function. An iterative algorithm is developed to calculate the parameter m_i, the flow chart of which is shown in Fig. 4.3. The calculation of m_i starts with $i = 1$, and m_i is then sequentially calculated. Finally, the program terminates when the condition $2m_i - 1 \geq M$ is satisfied, where $M = 2m - 1$.

After m_i is derived, other parameters can be derived if they can be expressed as a function of m_i. Fig. 4.2 shows the parameters required for GMMA. The parameters and their corresponding algorithms are defined as follows.

a. The number of symbol levels m_{si} for each sample set $y_{n,i}$. For the ith sample set, m_{si} can be computed if m_i is provided

$$m_{si} = m_i - m_{i-1} \tag{4.21}$$

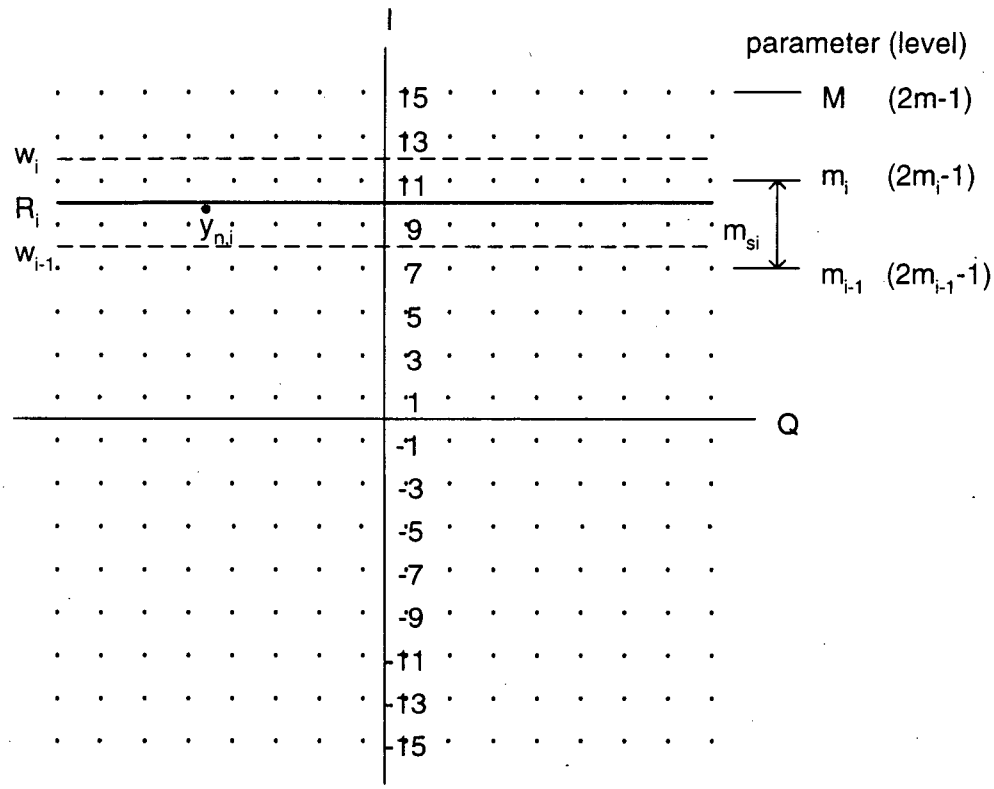


Figure 4.2: GMMA design parameters for 256-CAP

Fig. 4.2 shows the three parameters in a 256-point constellation.

b. The sample boundary w_i. The parameter w_i is defined as

$$w_i = 2m_i \tag{4.22}$$

w_i is the parameter required to describe the sample subset $y_{n.i}$. Note that w_i is chosen as above because odd number are used for the symbols. With such a definition and symbol levels which are odd integers, the sample boundary lies at equal distance between two symbol levels. When w_i is defined, the sample subset $y_{n.i}$ is described as

$$w_{i-1} \leq |y_{n,i}| \leq w_i \tag{4.23}$$

Consequently, the symbol subset $a_{n,i}$ is described as

$$w_{i-1} < |a_{n,i}| < w_i \tag{4.24}$$

Fig. 4.2 shows w_i and w_{i-1} as two dotted lines.

c. The constant R_i for subset $y_{n,i}$. When m_i is defined, the constant R_i can be easily calculated. Details can be found in [30]. Fig 4.2 shows the required R_i for sample set $y_{n,i}$ as a solid line.

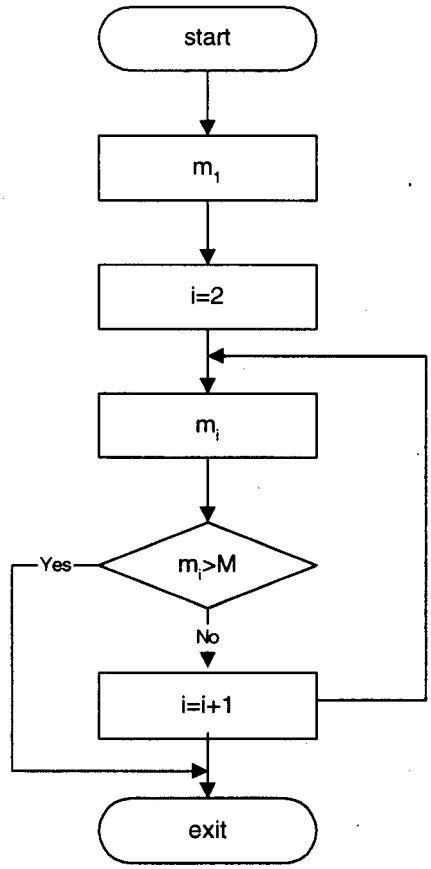


Figure 4.3: Flow-chart for the design of the GMMA parameters

The general algorithm description to calculate the system parameters for GMMA has been provided. The special case of 256-CAP is now discussed. In this example, the number of symbol levels is given by $m = 8$, the largest symbol level is $M = 15$, and the residual values of the cost function CF_{an} in dB is chosen to be $C_{dB} = 28$ dB (experimental). The number $C_{dB} = 28$ dB refers to the residual values of CF_{an} required by MMA for 64-CAP. C_{dB} is normally chosen experimentally.

1. The calculation of m_i starts with $i = 1$. When the index i is equal to one, the calculation of m_i is simple. The cost function $CF_{an,1}$ as a function of m_1 is given by

$$CF_{an,1} = E[(a_{n,1}^2 - R_1^2)^2] = \frac{16}{75}(12m_1^4 - 19m_1^2 + 7) \tag{4.25}$$

In this design, we assume $C = 28$ dB. Note that the constant C in logarithmic scale needs to be converted to linear scale. Setting $CF_{an,1} = C$, we obtain one positive solution. Note that normally the solution is not an integer, and an integer solution is obtained by rounding the number m_i. The same rule will be applied in the rest of the report to obtain positive integer solutions. In this case, we have $m_1 = 4$. For the data set $y_{n,1}$, the number of symbol level m_{s1} is given by

$$m_{si} = m_i \quad \rightarrow \quad m_1 = 4 \tag{4.26}$$

The data boundary w_1 is then computed as

$$w_1 = 2m_1 = 2 * 4 = 8 \tag{4.27}$$

For this value of w_1, the samples $y_{n,1}$ are described as

$$0 \leq |y_{n,1}| \leq 8 \tag{4.28}$$

Correspondingly, the symbols $a_{n,1}$ are restricted to

$$a_{n,1} = \{\pm 1, \pm 3, \pm 5, \pm 7\} \tag{4.29}$$

The constant R_1 can be computed from

$$R_1^2 = \frac{E[a_{n1}^4]}{E[a_{n1}^2]} = \frac{12m_1^2 - 7}{5} = 37 \rightarrow R_1 \approx 6 \tag{4.30}$$

For sample set $y_{n,1}$, the equalizer minimizes the dispersion around the constant $R_1 = 6$ when the equalizer's output samples satisfying $0 \leq |y_{n,1}| \leq 8$ are used during the filter adaptation.

2. The above described the derivation for $i = 1$. For $i \neq 1$, the cost function $CF_{an,i}$ is written as

$$CF_{an,i} = E[(a_{n,i}^2 - R_i^2)^2] = R_i^4 - E[a_{n,i}^4] \tag{4.31}$$

To compute $CF_{an,i}$, both R_i and $a_{n,i}$ must be determined as a function of m_i. It should be noted that the initial index of $a_{n,i}$ does not start with one. In order to use the results in [29] to compute the symbol moments, we rewrite $E[a_{n,i}^4]$ as follows

$$E[a_{n,i}^4] = \frac{1}{m_{si}}(\sum_{n=1}^{m_i} a_n^4 - \sum_{n=1}^{m_{i-1}} a_n^4) \tag{4.32}$$

where $m_{si} = m_i - m_{i-1}$. Then the expectation of $a_{n,i}$ becomes

$$E[a_{n,i}^4] = \frac{1}{15m_{si}}[m_i(48m_i^4 - 40m_i^2 + 7) - m_{i-1}(40m_{i-1}^4 - 40m_{i-1}^2 + 7)] \tag{4.33}$$

and the constant R_i is given by

$$R_i^2 = \frac{E[a_{n,i}^4]}{E[a_{n,i}^2]} = \frac{12}{5}(m_i^2 - m_{i-1}^2) \tag{4.34}$$

Substituting R_i^2 and $E[a_{n,i}^4]$ into the cost function $CF_{a,i}$, the following results:

$$\begin{aligned}
CF_{a,i} &= R_i^4 - E[a_{n,i}^4] && (4.35)\\
&= (\frac{12}{5}(m_i^2 - m_{i-1}^2))^2 - \frac{1}{15m_{si}}[m_i(48m_i^4 - 40m_i^2 + 7) && (4.36)\\
&\quad -m_{i-1}(40m_{i-1}^4 - 40m_{i-1}^2 + 7)] && (4.37)\\
&= \frac{64}{25}(m_i^4 - m_{i-1}^4) - \frac{368}{25}m_i^2 m_{i-1}^2 - \frac{16}{5}m_i m_{i-1}(m_i^2 - m_{i-1}^2) && (4.38)\\
&\quad +\frac{8}{3}(m_i^2 + m_i m_{i-1} + m_{i-1}^2) - \frac{7}{15} && (4.39)
\end{aligned}$$

In the iterative algorithm, m_{i-1} is known, and m_i is unknown. One variable is unknown for the equation $CF_{a,i} = C$, so that the equation can be uniquely solved. For $i = 2$ and $m_1 = 4$, we obtain the positive integer solution $m_2 = 6$. The following is the calculation of the number m_{si} for $i = 2$

$$m_{si} = m_i - m_{i-1} \quad \rightarrow \quad m_{s2} = m_2 - m_1 = 6 - 4 = 2 \tag{4.40}$$

The sample boundary is given by

$$w_i = 2m_{i2} = 2 * 6 = 12 \tag{4.41}$$

With $w_{i-1} = 8$, the sample subset $y_{n,2}$ is defined by

$$8 \leq |y_{n,2}| \leq 12 \tag{4.42}$$

For the symbol subset $a_{n,2}$, two symbol levels are included

$$a_{n,2} = \{\pm 9, \pm 11\} \tag{4.43}$$

With known m_i and m_{i-1}, the constant R_i can be computed. For $i = 2$, we have

$$R_2^2 = \frac{12}{5}(m_2^2 - m_1^2) = \frac{12}{5}(6^2 - 4^2) = 105 \quad \rightarrow \quad R_2 = 10.25 \tag{4.44}$$

For sample set $y_{n,2}$, the equalizer minimizes the dispersion around the constant $R_2 = 10.25$ when the samples are restricted to $8 \leq |y_{n,2}| \leq 12$.

3. In this step, the conditional equation $2m_i - 1 \geq M$ is tested. For $i = 2$ and $M = 15$, the condition is not satisfied since $2m_2 - 1 = 11 < M$. Thus, the index counter is incremented with $i = i + 1 = 3$. The program returns to step 2. Repeating the calculation in step 2 for $i = 3$, the following values are obtained: $m_3 = 8$, $m_{s3} = 2$, $w_3 = 16$ and $R_3 = 14.17$. Therefore, the samples $y_{n,3}$ are defined as

$$12 \leq |y_{n,3}| \leq \infty \tag{4.45}$$

and two symbol levels are included in $a_{n,3}$

$$a_{n,3} = \{\pm 13, \pm 15\} \tag{4.46}$$

For the sample set $y_{n,3}$, the equalizer minimizes the dispersion around $R_3 = 14.17$ when samples $12 \leq |y_{n,3}| \leq \infty$ are used during the tap updating.

4. The conditional equation $m_i \geq M$ is satisfied with $m_3 = 8$ since $2m_i - 1 = 15 = M$, so that the iterative algorithm for 256-CAP is terminated for $i = 3$.

This completes the parameter calculation of GMMA for 256-CAP. To satisfy the equal cost function minimization, a total of three sample subsets is required, as shown in Fig. 4.2. In this figure, the three sample subsets are divided by the dotted lines w_i, and three moduli R_i are represented by solid lines. The equalizer minimizes the dispersion at the equalizer's output samples with respect to three moduli, respectively, during the filter adaptation.

4.4 Examples

Table 4.1 shows the parameters calculated for 256-CAP, where $y_{n,1}$ is for MMA, $y_{n,1}$ and $y_{n,4}$ are for half-constellation MMA, and $y_{n,1}$, $y_{n,2}$ and $y_{n,3}$ are for GMMA. In fact, three different

Constant/a_n	$1 \sim 7$	$9 \sim 11$	$13 \sim 15$	$9 \sim 15$	$1 \sim 15$
$y_{n,i}$	$y_{n,1}$	$y_{n,2}$	$y_{n,3}$	$y_{n,4}$	$y_{n,5}$
CF	27.7	26	29	35	40
m_{wi}	4	2	2	4	8
R	6	10.35	14.2	13	12.34
$w_i \sim w_{i-1}$	$1 \sim 8$	$8 \sim 12$	$12 \sim \infty$	$8 \sim \infty$	$1 \sim \infty$

Table 4.1: Parameters for 256-CAP

algorithms are characterized in Table 4.1, the standard MMA, the half-constellation MMA, and the GMMA. For MMA with sample set $y_{n,5}$, in the last column of Table 4.1, the residual value of the cost function $CF_{a,n}$ is very high. About 40 dB of residual value exists when the equalizer converges. When the sample set y_n is equally split between subsets $y_{n,1}$ and $y_{n,4}$, the residual value for the second half constellation is about 8 dB larger than that of the first half. It means that equal cost function minimization is not achieved with a subset selection that simply splits the signal constellation into two subsets having the same number of symbols. Finally, equivalent cost function minimization is achieved by the GMMA with equal cost function minimization. As shown in Table 4.1, when three data subsets are used as $y_{n,1}$, $y_{n,2}$ and $y_{n,3}$, the residual values of the cost function $CF_{an,i}$ are reduced to be approximately the same.

Summary: GMMA is proposed as an extension of MMA by using multiple moduli along each dimension. The cost function analysis showed that the residual values of the cost function are nonzero, and these values grow when the number of symbol levels increases. We need to reduce those minimum values if we want to improve the performance of the eye opening. This is realized

by using multiple moduli along each dimension of symbols. By doing this, we introduce sample subsets into the algorithm. The residual values of the cost functions can then be minimized in each subset. In this case, each constant R can efficiently represent the knowledge of the statistics of the symbols in the corresponding subset.

In this chapter, the use of multiple moduli is demonstrated by the design given for 256-CAP. Because sample subsets are not able to be determined at the beginning, R cannot be simply derived. The calculation of multiple R is provided by a sequential computing algorithm. The analysis shows that by applying three constants R_i to a 256-CAP, the cost function can be minimized to similar small values. This greatly facilitates the opening of the eye.

Chapter 5

Windowed MMA (WMMA)

With MMA, minimum values of the cost functions are generally nonzero and the values increase when the number of symbol levels increases. With GMMA, reduction of the minimum value of the cost function is realized by using multiple moduli along each dimension. Windowed algorithms for MMA are proposed as another approach to realize a reduction of this minimum value. The study shows that errors contribute different effects to filter adaptation depending on how close the samples are distributed around the constant R. The convergence rate can be improved by using only part of data. Two algorithms will be proposed as applications of windowed algorithms for MMA with improved convergence rate [30].

5.1 Algorithm Structure

It was shown in the previous section that a cost function usually does not converge to zero with blind algorithms except for 4-CAP. Both the eye diagram opening and convergence rate are affected by residual values of the cost function, or more accurately, by erroneous quantities used in the cost function. Let Δe denote the difference between the error $e_{r,n}(CF)$ and the error $e_{r,n}(LMS)$, as shown in Fig. 4.1. The convergence rate of an equalizer is a decreasing function of the value of Δe. Fig. 4.1 shows that the values of Δe vary according to its distribution. If Δe can be minimized by some constraints, the equalizer's convergence can be improved. A new algorithm is required so that the cost function can minimize the error difference Δe.

A modified version of MMA, called windowed MMA, is proposed as a new algorithm with improved convergence properties. In this algorithm, the cost function is designed in such a way that the error difference Δe can be minimized. Further study of Fig. 4.1 shows that Δe has different effects on the cost function depending on the absolute value of the distance $|y_n| - R$, or the second-order quantity $y_n^2 - R^2$. Fig. 4.1 shows clearly that the error difference Δe is proportional to the distance $||y_n| - R|$. Therefore, a new algorithm is developed so that filter tap updating needs to be done only with those data which have relatively small values of $||y_n| - R|$.

WMMA is a modified version of MMA, except that a sample window is used during adaptation. The WMMA scheme is illustrated in Fig. 5.1 where the sample window is defined by two dotted lines along each dimension. With WMMA, only some of the samples $y_{n,w}$ are used during filter adaptation, whereas with MMA all the samples are used. Fig. 5.1 shows the window parameters defined with half-constellation WMMA for 64-CAP. When $m_w = 4$ is used as the boundary line of the sample window, the samples $y_{n,w}$ are given by $|y_n| > m_w$, or $|y_n| > 4$. Consequently, the new constellation used for adaptation includes symbols $a_{n,w} = \{\pm 5, \pm 7\}$. Fig. 5.1 also shows the new constants R represented by two solid lines, and $R = 6.4$ is required for the equalizer to converge to the constellation with symbols $a_{n,w} = \{\pm 5, \pm 7\}$.

Two applications of WMMA are the half-constellation WMMA, and the edge-point WMMA. The two algorithms use the same cost function, except for the different definitions of the sample windows. Because different sets of samples y_n are used during filter adaptation, different results can be achieved in terms of cost function optimization.

5.2 Half-Constellation WMMA

The half-constellation WMMA is proposed as the first implementation of the WMMA concept, and the schematic diagram of this algorithm for 64-CAP is illustrated in Fig. 5.1, which shows the sample window. In the case of the in-phase dimension the samples y_n are divided into two sets by the window boundaries m_w with $|y_n| \leq m_w$ and $|y_{n,w}| > m_w$. With new samples $y_{n,w}$, the cost function CF is redefined as

$$CF = E[(y_{n,w}^2 - R_w^2)^2] \tag{5.1}$$

Note that the constant R is changed to R_w because only a subset of signals $y_{n,w}$ is used to converge the equalizer. The equalizer's taps are updated with samples $y_{n.w}$ and remain unchanged with samples y_n. For the half-constellation WMMA, the window boundary m_w is defined as

$$m_w = m \tag{5.2}$$

where m indicates the number of symbol levels, and the magnitude of the highest symbol level is given by $2m - 1$. The window boundary m_w is defined in such a way that the same number of inner-point and outer-point symbols a_n are included around the constant R. In other words, we can say that the samples $y_{n,w}$ used to update the taps are symmetrically distributed on both sides of R. It is called half-constellation WMMA because the number of partial symbols $a_{n,w}$ that construct the new constellation is half the number of the original symbols a_n. For the half-constellation WMMA, the constant R_w needs to be evaluated with respect to symbols $a_{n,w}$. With the samples $y_{n,w}$, the cost function CF_w now converges to the symbols $a_{n,w}$

$$CF = E[(y_{n,w}^2 - R_w^2)^2] \quad \rightarrow \quad CF = E[(a_{n,w}^2 - R_w^2)^2] \tag{5.3}$$

Then the constant R_w is computed as

$$R_w = \frac{E[a_{n,w}^4]}{E[a_{n,w}^2]} \tag{5.4}$$

Note that the initial index of $a_{n,w}$ does not start with one. In order to use the results derived for the symbols a_n in [30], we need to derive the moments for the symbols $a_{n,w}$ with arbitrary initial index. An example is given for the calculation of $E[a_{n^2,w}]$. The second-order expectation is rewritten as

$$E[a_{n,w}^2] = \frac{1}{w}(\sum_{n=1}^{M} a_n^2 - \sum_{n=1}^{M_w} a_n^2) \tag{5.5}$$

The parameter w denotes the number of symbol levels required for the constellation including $a_{n,w}$. For the half-constellation WMMA, we simply have

$$w = \frac{1}{2}m \tag{5.6}$$

Eq. 5.6 means that half the number of the original symbol levels m is required. The parameter M denotes the number of the largest symbol levels and the parameter M_w denotes the number of

symbol levels below the window boundary m_w. The parameters are given by

$$M = 2m - 1 \quad \text{and} \quad M_w = m - 1 \tag{5.7}$$

The constant R_w can then be calculated as

$$\begin{aligned}
R_w^2 &= \frac{E[a_{n,w}^4]}{E[a_{n,w}^2]} \\
&= \frac{\frac{1}{w}[\sum_{n=1}^{M} a_n^4 - [\sum_{n=1}^{M_w} a_n^4]}{\frac{1}{w}[\sum_{n=1}^{M} a_n^2 - [\sum_{n=1}^{M_w} a_n^2]} \\
&= \frac{\frac{2}{m}[\sum_{n=1}^{m} a_n^4 - [\sum_{n=1}^{w} a_n^4]}{\frac{2}{m}[\sum_{n=1}^{m} a_n^2 - [\sum_{n=1}^{w} a_n^2]} \\
&= \frac{\frac{m}{15}(48m^4 - 40m^2 + 7) - \frac{m}{30}(3m^4 - 10m^2 + 7)}{\frac{m}{3}(4m^2 - 1) - \frac{m}{6}(m^2 - 1)} \\
&= \frac{m(\frac{31}{10}m^4 - \frac{7}{3}m^2 + \frac{7}{30})}{\frac{m}{6}(7m^2 - 1)} \\
&= \frac{93m^4 - 70m^2 + 7}{35m^2 - 5}
\end{aligned}$$

Values of the constant R for MMA and the constant R_w for WMMA are provided in Table 5.1. We

Parameter/CAP	16-CAP	64-CAP	256-CAP	1024-CAP
m	2	4	8	16
R	2.86	6.08	12.34	24.76
m_w	1	2	4	8
R_w	3	6.4	12.97	26.05

Table 5.1: Constant R for half-constellation WMMA

observe in Table 5.1 that the values of R_w are always larger than those of R. It means that the constant R is smaller when the initial symbol starts from one.

Next we compute the residual values of the cost function for the half-constellation WMMA to show how much reduction can be obtained. The simplified expression for the cost function CF_{an}

was derived in Section 1. It changes to the following with R_w and $E[a_{n,w}^4]$

$$CF_{an} = R^4 - E[a_n^4] \quad \rightarrow \quad CF_{a_n} = R_w^4 - E[a_{n,w}^4] \tag{5.8}$$

Replacing R_w^2 with

$$R_w^2 = \frac{93m^4 - 70m^2 + 7}{35m^2 - 5} \tag{5.9}$$

and replacing $E[a_{n,w}^4]$ with

$$E[a_{n,w}^4] = \frac{2(93m^4 - 70m^2 + 7)}{m} \tag{5.10}$$

the cost function $CF_{a,w}$ for the half-constellation WMMA is then obtained as

$$\begin{aligned} CF_{an} &= R_w^4 - E[(a_{n,w}^2 - R_w^2)^2] \\ &= R_w^4 - E[a_{n,w}^4] \\ &= \frac{(93m^4 - 70m^2 + 7)^2}{(35m^2 - 5)^2} - 2 * \frac{93m^4 - 70m^2 + 7}{m} \\ &= \frac{2}{75} * \frac{(93m^4 - 70m^2 + 7)(m + 2)(m - 2)(17m^2 - 2)}{7m^2 - 1} \end{aligned}$$

Setting $CF = 0$, we obtain one positive integer solution, which is $m = 2$ for 16-CAP. The comparison of residual values of cost functions for MMA and half-constellation WMMA is provided in Table 5.2. It can be seen in this Table that the cost function for 16-CAP becomes zero, and that a

Cost function/CAP	16-CAP	64-CAP	256-CAP	1024-CAP
m	2	4	8	16
CF_{mma} (dB)	14.2	27.7	40	52
CF_{wmma} (dB)	0	22	35	47
ΔCF	14.2	5.7	4.95	4.79

Table 5.2: Comparison of cost functions

reduction of about 5 dB can be obtained for other CAP systems. Thus, by using WMMA, the cost function $CF_{a,n}$ becomes optimum for 16-CAP. For other CAP systems, the cost function $CF_{a,n}$ is

reduced. Reduction of the residual values of the cost function $CF_{a,n}$ leads to improved reliability and convergence rate of blind equalizers.

5.3 Edge-Point WMMA

Edge-point WMMA is proposed as the second application of WMMA. The cost function for half-constellation WMMA in (5.1) can be directly applied to edge-point WMMA, except that a modification is made for the sample window parameters. The window parameters are illustrated in Fig. 5.2, where the window boundary m_w is defined as

$$m_w = 2(m-1) \tag{5.11}$$

With such a definition, the symbols $a_{n,w}$ are given by a_{2m-1}. The symbols used for filter adaptation are only those that have the largest values. Fig. 5.2 shows that these symbols are geometrically located at the edge of the original constellation. Because only one symbol level is involved, the calculation of R_w is simply given by

$$R_w = a_{n,w} = 2m-1 \tag{5.12}$$

The above equation yields the following equality

$$R_w^2 = E[a_{n,w}^2] \tag{5.13}$$

Eq. 5.13 further leads to the result

$$CF_{a,w} = E[(a_{n,w}^2 - R_w^2)^2] \quad \rightarrow \quad 0 \tag{5.14}$$

Eq. 5.14 shows that zero value can be achieved with such a cost function. That is, with edge-point WMMA, the cost function becomes optimum for any CAP system.

The edge-point and half-constellation MMA are basically the same except that they use different window parameters. However, the difference in parameter results in different performance. Theoretically, optimum convergence can only be achieved for 16-CAP with half-constellation WMMA, and can be achieved for any CAP application with edge-point WMMA under conditions to be discussed later.

The choice of design parameters of edge-point constellation is simple and easy to implement. However, expected performance cannot be achieved for high level CAP applications because other factors also affect convergence, such as the lack of data samples y_n.

5.4 Filter Adaptation

Two WMMA algorithms were proposed in the previous section. We now derive the algorithms for updating the tap coefficients. For simplicity, the previous analysis for WMMA was only given for the in-phase dimension. The complete two-dimensional cost function for WMMA is given by

$$CF = [(y_{n,w}^2 - R_w^2)^2 + (\tilde{y}_{n,w}^2 - R_w^2)^2] \tag{5.15}$$

The cost functions can be applied to both half-constellation and edge-point WMMAs by using different definitions for $y_{n,w}$, as shown in Section 2. The gradients of the cost function in (5.15) with respect to the tap vectors $\mathbf{c}_n$ and $\mathbf{d}_n$ are equal to

$$\nabla_c = (y_{n,w}^2 - R_w^2)y_{n,w}\mathbf{r}_n \qquad \nabla_d = (\tilde{y}_{n,w}^2 - R_w^2)\tilde{y}_{n,w}\mathbf{r}_n \tag{5.16}$$

The filter's taps are then updated in a stochastic fashion in the opposite direction of the gradient

$$\mathbf{c}_{n+1} = \mathbf{c}_n - \mu(y_{n,w}^2 - R_w^2)y_{n,w}\mathbf{r}_n \tag{5.17}$$

$$\mathbf{d}_{n+1} = \mathbf{d}_n - \mu(\tilde{y}_{n,w}^2 - R_w^2)\tilde{y}_{n,w}\mathbf{r}_n \tag{5.18}$$

Note that $y_{n,w}$ and R_w are different for the two versions of WMMA.

In the implementation of the algorithms, the samples $y_{n,w}$ are normally calculated by using a comparator, or a look-up table. Alternatively, a nonlinear function $f(\cdot)$ is proposed to determine partial samples $y_{n,w}$. The function $f(\cdot)$ is defined as

$$f(y_n) = \frac{1}{2}[1 + sgn(y_n^2 - m_w^2)] \tag{5.19}$$

$$f(\tilde{y}_n) = \frac{1}{2}[1 + sgn(\tilde{y}_n^2 - m_w^2)] \tag{5.20}$$

$$\tag{5.21}$$

So that

$$f(y_n) = \begin{cases} 0 & \text{if } |y| \leq m_w \\ 1 & \text{otherwise} \end{cases} \qquad f(\tilde{y}_n) = \begin{cases} 0 & \text{if } |\tilde{y}| \leq m_w \\ 1 & \text{otherwise} \end{cases}$$

where $m_w = m$ for half-constellation WMMA, and $m_w = 2(m-1)$ for edge-point WMMA. The use of the nonlinear equation $f(.)$ gives the following relation

$$f(y_n) = y_{n,w} \qquad f(\tilde{y}_n) = \tilde{y}_{n,w} \tag{5.22}$$

The cost function CF can be rewritten as

$$CF = [f(y_n)(y_n^2 - R_w^2)^2 + f(\tilde{y}_n)(\tilde{y}_n^2 - R_w^2)^2] \tag{5.23}$$

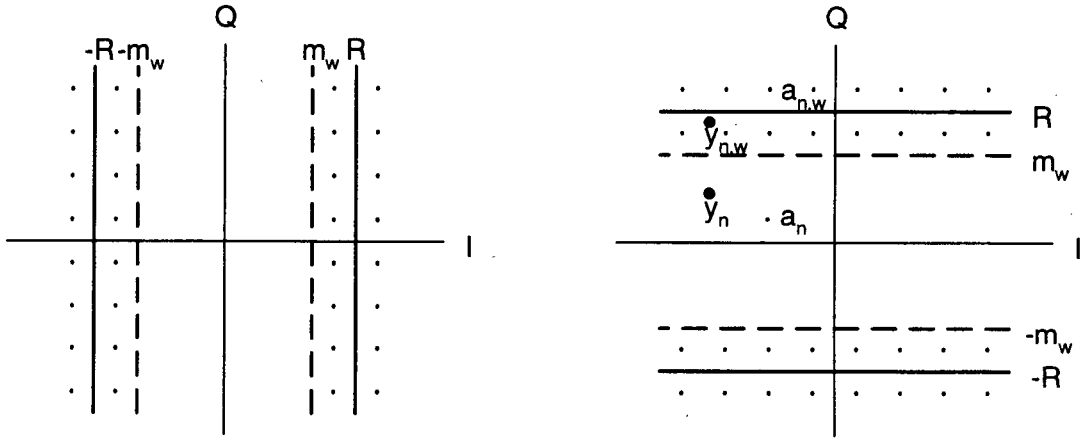
and the corresponding tap updating algorithm becomes

$$\mathbf{c}_{n+1} = \mathbf{c}_n - \mu f(y_n)(y_n^2 - R_w^2) y_n \mathbf{r}_n \tag{5.24}$$

$$\mathbf{d}_{n+1} = \mathbf{d}_n - \mu f(\tilde{y}_n)(\tilde{y}_n^2 - R_w^2) \tilde{y}_n \mathbf{r}_n \tag{5.25}$$

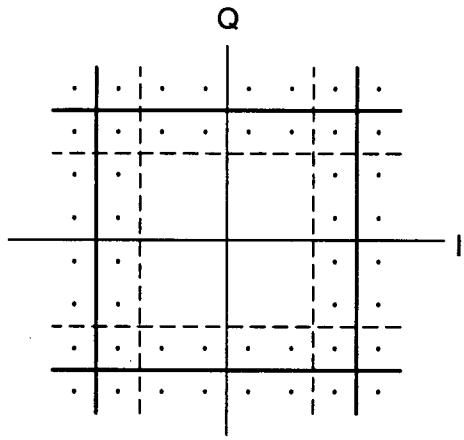
In the case of the in-phase dimension, both (5.24) and (5.17) can be used to do filter adaptation.

Summary: The half-constellation MMA and edge-point MMA are proposed as improved MMA algorithms. Both algorithms provide some improvement in cost function minimization. In theory, cost function optimization can be achieved for any application. However, filter adaptation cannot be done with insufficient samples. So the windowed algorithms are limited to some specific applications. In particular, WMMA is attractive to some applications with a medium size number of symbol levels. Even though only partial data are used, a better convergence performance can be achieved than with the full algorithm implementation because the residual value of the WMMA cost function is smaller than that of MMA and it leads to reduction of the tap updating fluctuation.



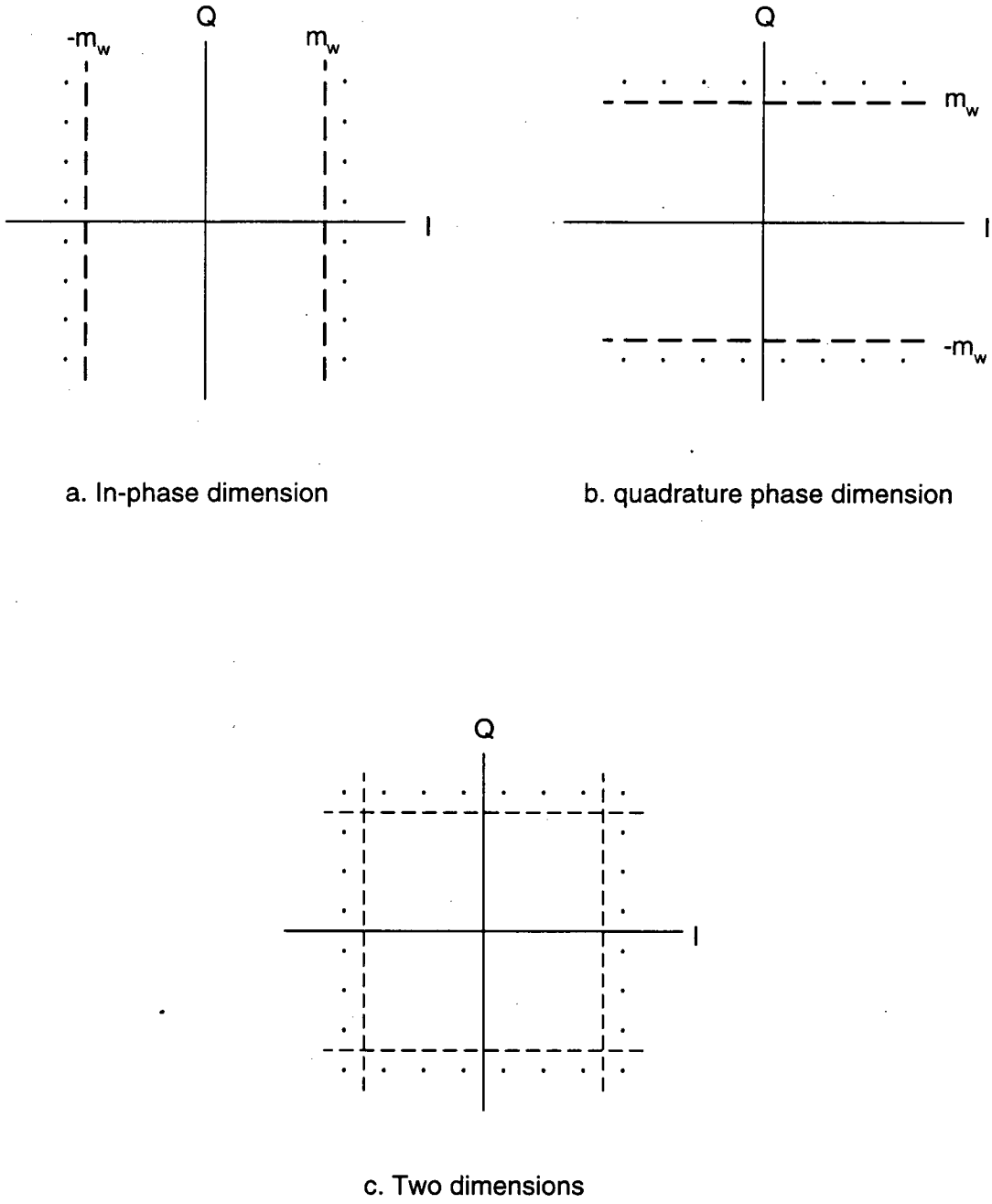
a. In-phase dimension

b. quadrature phase dimension



c. Two dimensions

Figure 5.1: Half-constellation WMMA for 64-CAP



a. In-phase dimension

b. quadrature phase dimension

c. Two dimensions

Figure 5.2: Edge-point WMMA for 64-CAP

Chapter 6

Computer Simulations and Laboratory Experiments

In Chapter 3 through Chapter 5, we have presented a detailed study of MMA and its generalized algorithms. The improved convergence performance of MMA is demonstrated in this chapter through computer simulations and laboratory experiments.

6.1 Simulation Environment

In chapter 2, we described in detail the receiver structures required for blind equalization. Amongst the possible alternatives, we chose a receiver using a phase-splitting equalizer of the type that is used for the 64-CAP ATM LAN standard. [1]. This standard specifies square-root raised cosine shaping filters with 15% excess bandwidth. The symbol rate for the ATM standard is $1/T = 25.92$ Mbaud, and a $T/3$ tap spacing is used for the FSLE, so that the sampling rate of the A/D is 77.76 MHz. The two shaping filters used at the transmitter form a Hilbert pair. That is, they have the same amplitude characteristics, and phase characteristics that differ by 90 degrees. With the above parameters, we obtain a data rate of 155.52 Mb/s for 64-CAP, and 207.36 Mb/s for 256-CAP. In laboratory experiments, signals are transmitted over an 100 meter unshielded twisted pair (UTP) category 3 cable. In the computer simulations, we use the equivalent channel models provided in

[18].

At the receiver, we chose the phase-splitting filter as the equalizer's configuration. In the ATM LAN standard for 64-CAP, an equalizer span of $16T$ is required in [18]. So that a total number of 48 taps is used for each filter of the equalizer. This number of taps has been used in all the simulations presented here for 64-CAP and 256-CAP. Most of the simulations use 64-CAP and 256-CAP, but CAP systems using other signal constellations have been studied.

In typical ATM LAN networks, the main noise in the system is near-end crosstalk (NEXT). Cancellation of NEXT can be efficiently done with NEXT cancelers or NEXT equalizers [18]. In this thesis, in order to emphasis the study of blind algorithms, we are only concerned with simulations operating without noise, which is justified by the fact that ISI is the dominant interference during initial training.

Reliable convergence includes two criteria. One is the quality of the eye diagram opening, and the other is the reduction of wrong solutions, particularly of diagonal solutions for the phase splitting equalizer. The reduction of the probability of creating wrong solutions is discussed in related work by the authors [30] and is not covered in this thesis.

6.2 Computer Experiments

The performance of MMA and its generalized algorithms have been tested through computer simulations using the above-mentioned design parameters.

6.2.1 MMA

We first test MMA for the 64-CAP application. It is particularly interesting to compare the convergence performances of MMA and RCA.

Fig. 6.1 shows the typical MSE learning curves for RCA and MMA in multiple simulation runs. Note that the sudden desending with those curves show the switching from blind equalization to the LMS algorithm. It is obvious from the figure that the equalizer converges faster with MMA than with RCA. The equalizer's convergence time with MMA is half that needed for RCA. Note that the same step size $\mu = 0.0001$ is used for both algorithms. MMA can converge faster than

RCA because it uses higher order statistics of the signals [11].

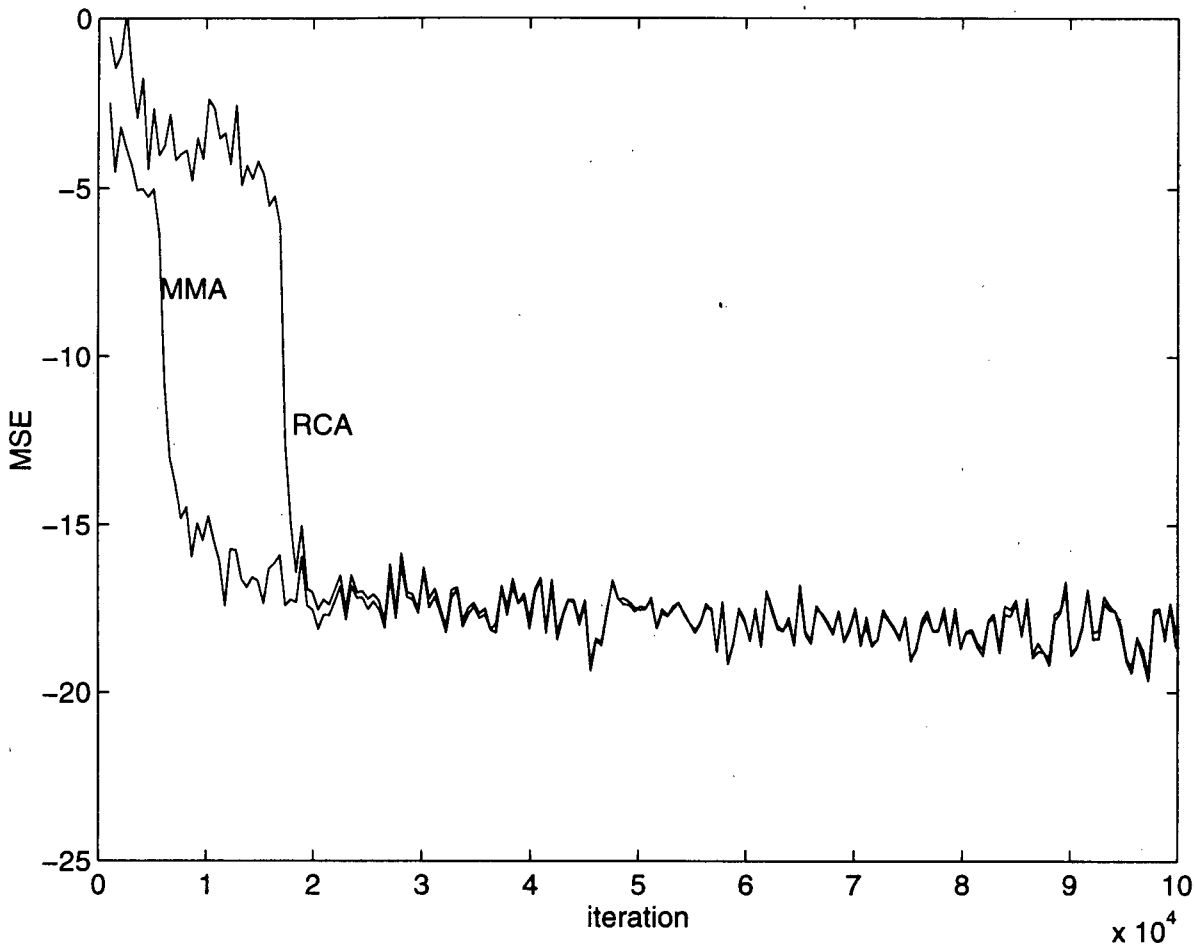


Figure 6.1: MSE learning curves for various blind algorithms used for 64-CAP

However, it should be pointed out that recent experimental results seem to show that this advantage of MMA may not be significant for small signal constellations, such as 4-CAP and 16-CAP, which are used in other types of applications.

The implementation of the MMA algorithm is more complicated than that of RCA. The tap updating algorithms for RCA and MMA are given by the following equations

$$\mathbf{c}_{n+1} = \mathbf{c}_n - \mu[y_n - Rsgn(y_n)]\mathbf{r}_n \tag{6.1}$$

$$\mathbf{c}_{n+1} = \mathbf{c}_n - \mu(y_n^2 - R^2)y_n\mathbf{r}_n \tag{6.2}$$

where (6.1) is used for RCA and (6.2) is used for MMA. The above tap updating algorithms apply to the in-phase dimension, and similar equations apply to the quadrature dimension. Eq. (6.2) requires higher-order calculations than (6.1). However, the computational cost can be significantly reduced for MMA if a power-of-two multiplier is used for tap updating [6]. The use of MMA not only provides a good convergence rate, but it also gives a good immunity against wrong convergence to diagonal solutions. Comparing to RCA, MMA reduces the probability of diagonal solutions by about 90% with our computer simulations.

For blind start-up, a satisfactory convergence time for the applications considered here is 1 to 2 seconds. For the 16-CAP 51.84 Mb/s system, the symbol rate 1/T is equal to 12.96 Mbaud, so that the symbol period T is equal to $T = \frac{1}{12.96 \times 10^6}$ second. Assuming that each tap updating iteration is done in 500 symbol periods (500 T),which is easily done in VLSI, and assuming that is takes 30000 iterations to blindly converge the equalizer, we get a convergence time τ equal to

$$\tau = 30000 \times \frac{500}{12.96 \times 10^6} \approx 1.2 second \tag{6.3}$$

which is perfertly acceptable for the applications considered.

6.2.2 GMMA

The important feature of MMA is that multiple moduli are employed, whereas only one modulus is used for RCA and CMA. With MMA, the use of multiple moduli is especially useful for nonsquare constellations and constellations with a high number of symbol levels. In this chapter, we focus the discussion on how multiple moduli can be applied to high symbol level applications, in which case MMA is called generalized MMA (GMMA).

The convergence performance of GMMA is tested for the 256-CAP application. In the previous sections, we suggested to use multiple moduli for MMA in order to reduce J_{ex}, thus improving the eye opening. The parameters in Table 6.1 show that the residual values of the cost functions are changed when different MMA algorithms are used for the equalizer. Fig. 6.2 shows a comparison between the evolution of the cost functions for MMA and GMMA. The first plot, curve 1, is obtained for standard MMA. Notice that this curve has a small dynamic range and a large residual value. By using MMA, we obtain only about 2 dB of change, and J_{ex} is about 40 dB high for 256-CAP.

Fig. 6.3 shows the converged signal constellation with MMA. The poor performance of the eye opening is mainly caused by the small dynamics and high J_{ex} of the MMA cost function.

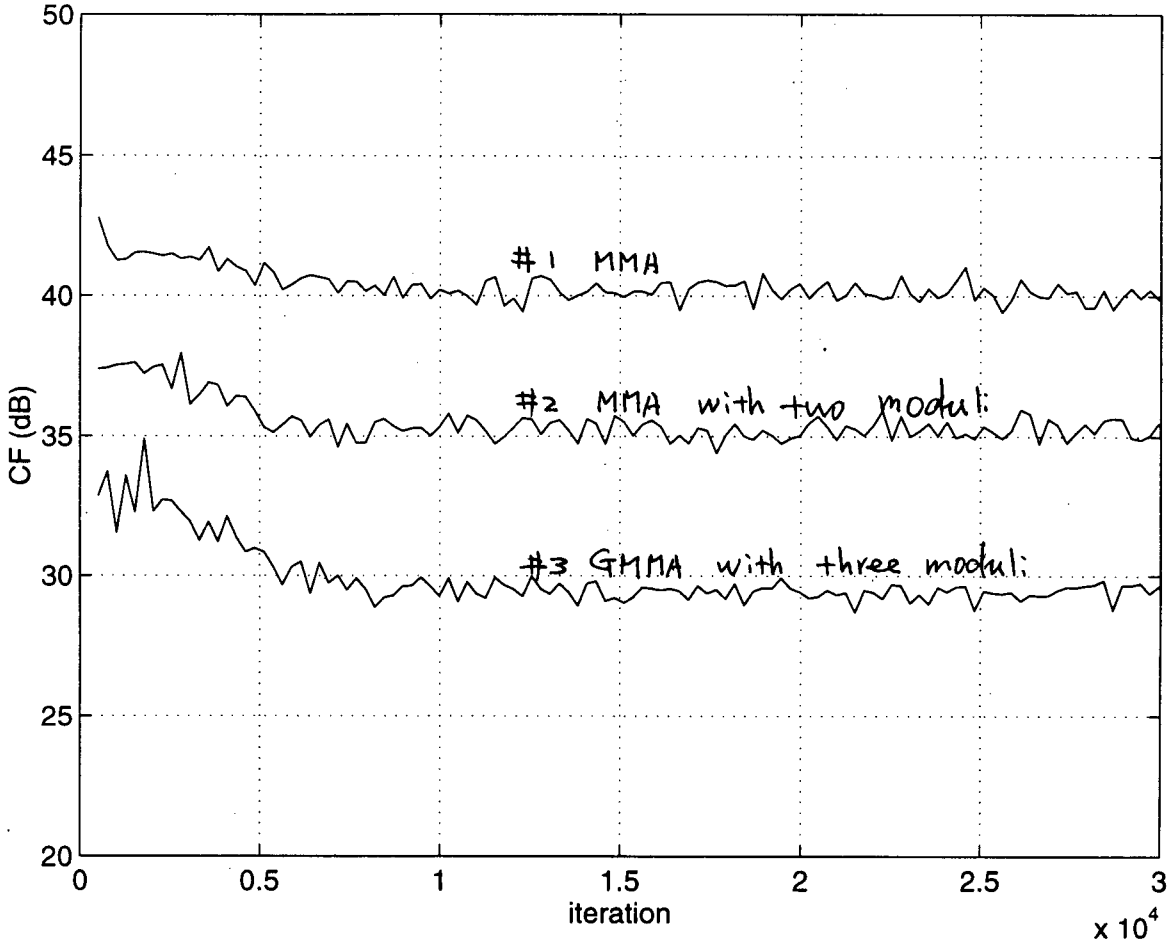


Figure 6.2: Cost function evolution for MMA and GMMA when used with 256-CAP

Curve 2 in Fig. 6.2 is obtained when multiple moduli are used for MMA. In this test, we use multiple moduli but not as previously specified for GMMA. In this test, we just equally split the output samples. We see from Table 6.1 that the residual values of the cost functions are equal to 27.7 dB and 35 dB if the samples are equally split in subsets $y_{n,2}$ and $y_{n,3}$. Notice that the use of equally divided samples does not lead to the same residual values for each sample subset. The maximum J_{ex} is only reduced by 5 dB, and the dynamics of the cost function is not improved very

much.

Curve 3 in Fig. 6.2 corresponds to GMMA for which the equal energy principle is applied to define the various moduli. In this test, the output samples are divided into three parts, for which the residual values of the cost function are equal to 27.7 dB, 26 dB, and 29 dB, as shown in Table 6.1, for subsets $y_{n,2}$, $y_{n,4}$ and $y_{n,5}$. Curve 3 in Fig. 6.2 shows the evolution of the cost function for the subset $a_n = \{\pm 13, \pm 15\}$, which has the largest residual cost function.

Table 6.1 shows cost function computations, and Fig. 6.2 shows computer simulation results. Notice that there is a very good match between the theoretical and computer simulation results.

Constant/a_n	$1 \sim 15$	$1 \sim 7$	$9 \sim 15$	$9 \sim 11$	$13 \sim 15$
$y_{n,i}$	$y_{n,1}$	$y_{n,2}$	$y_{n,3}$	$y_{n,4}$	$y_{n,5}$
CF	40	27.7	35	26	29
m_{wi}	8	4	4	2	2
R	12.34	6	13	10.35	14.2
$w_i \sim w_{i-1}$	$1 \sim \infty$	$1 \sim 8$	$8 \sim \infty$	$8 \sim 12$	$12 \sim \infty$

Table 6.1: Parameters for 256-CAP

Fig. 6.3 shows the signal constellation obtained with MMA, and Fig. 6.4 shows the signal constellation after converging with GMMA.Clearly, a better eye opening can be obtained by using GMMA rather than MMA. However, GMMA can only be applied to signal constellations with a number of symbols smaller or equal to 256. Otherwise, there are too many subsets of symbols which include one or two symbols only. Fig. 6.2 confirms the result stated in Chapter 4, which says that the best blind convergence is obtained with cost functions which have the smallest residual J_{ex} and the largest dynamics during convergence.

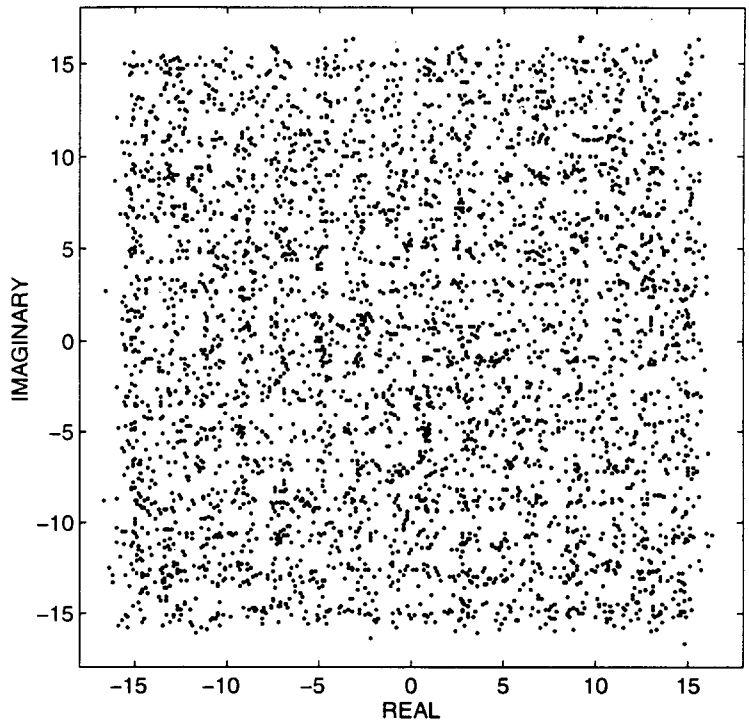


Figure 6.3: After convergence with MMA

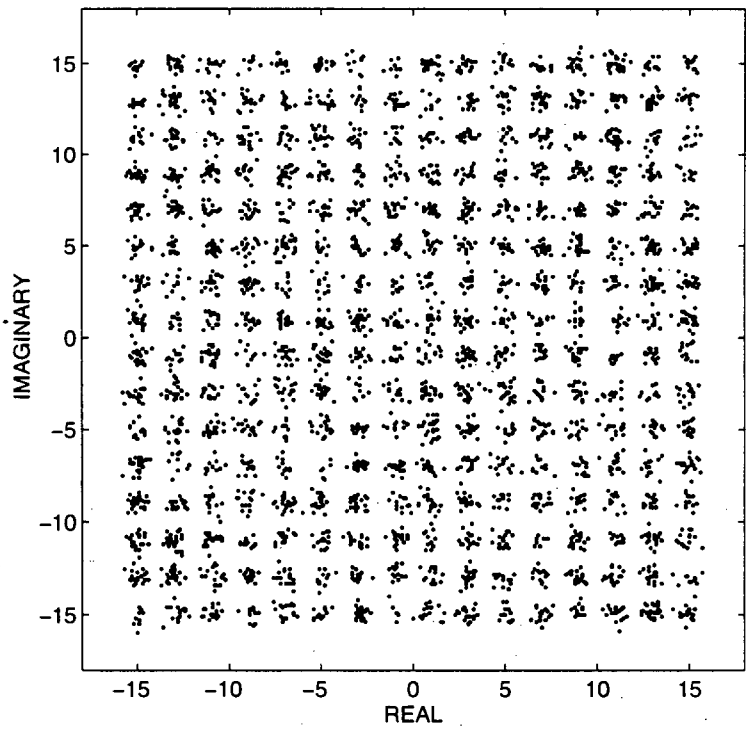


Figure 6.4: After convergence with optimized GMMA

6.2.3 WMMA

The use of multiple moduli is one way of reducing the residual values of the cost functions. In Chapter 5, we proposed WMMA as an alternative approach to improve cost function reduction. In this thesis, the performance of WMMA is tested for the 64-CAP application.

In contrast to GMMA, WMMA only uses one constant R, and cost function reduction is obtained by using a windowing technique. That is, we update the filters with only those samples which provide minimum residual values of the cost function. We have proposed two windowing algorithms, called the half-constellation algorithm and the edge-point constellation algorithm. The simulations will be analyzed with respect to the cost function and the MSE.

Evolution of the Cost Function

Computer simulation results for WMMA are shown in Fig. 6.5. This figure shows the evolution of the cost function during convergence for various WMMA algorithms. Curve 1 corresponds to the MMA cost function, and is given as a reference. As seen in Fig. 6.5, MMA can provide about 2 dB of variation for the cost function, which causes slow filter convergence.

The first variation of WMMA is the half-constellation WMMA, in which half of the samples are used during filter adaptation as shown in Fig. 5.1. Curve 2 in Fig. 6.5 gives the evolution of the cost function for the half-constellation WMMA. Notice the improved convergence in terms of reduced residual value and increased dynamic range of the cost function. For the 64-CAP application with the half-constellation WMMA, we computed $J_{ex} = 22$ dB, as shown in Table 5.2. Similar values are obtained through computer simulations, as shown in Fig. 6.5.

Curve 3 in Fig. 6.5 corresponds to the edge-point WMMA, in which only edge-point samples are used during filter adaptation. In theory, we can obtain an optimal solution by using the edge-point WMMA. In practice, the residual value of the cost function is not zero, but very small as shown by curve 3 in Fig. 6.5.

We give in Table 5.2 the theoretical calculations of J_{ex} for 64-CAP. Notice that the calculation gives $J_{ex} = 27.7$ dB for MMA, and $J_{ex} = 22$ dB for the half-constellation WMMA. The theoretical value of J_{ex} is equal to zero for the edge-point WMMA. As was the case for GMMA, we see that

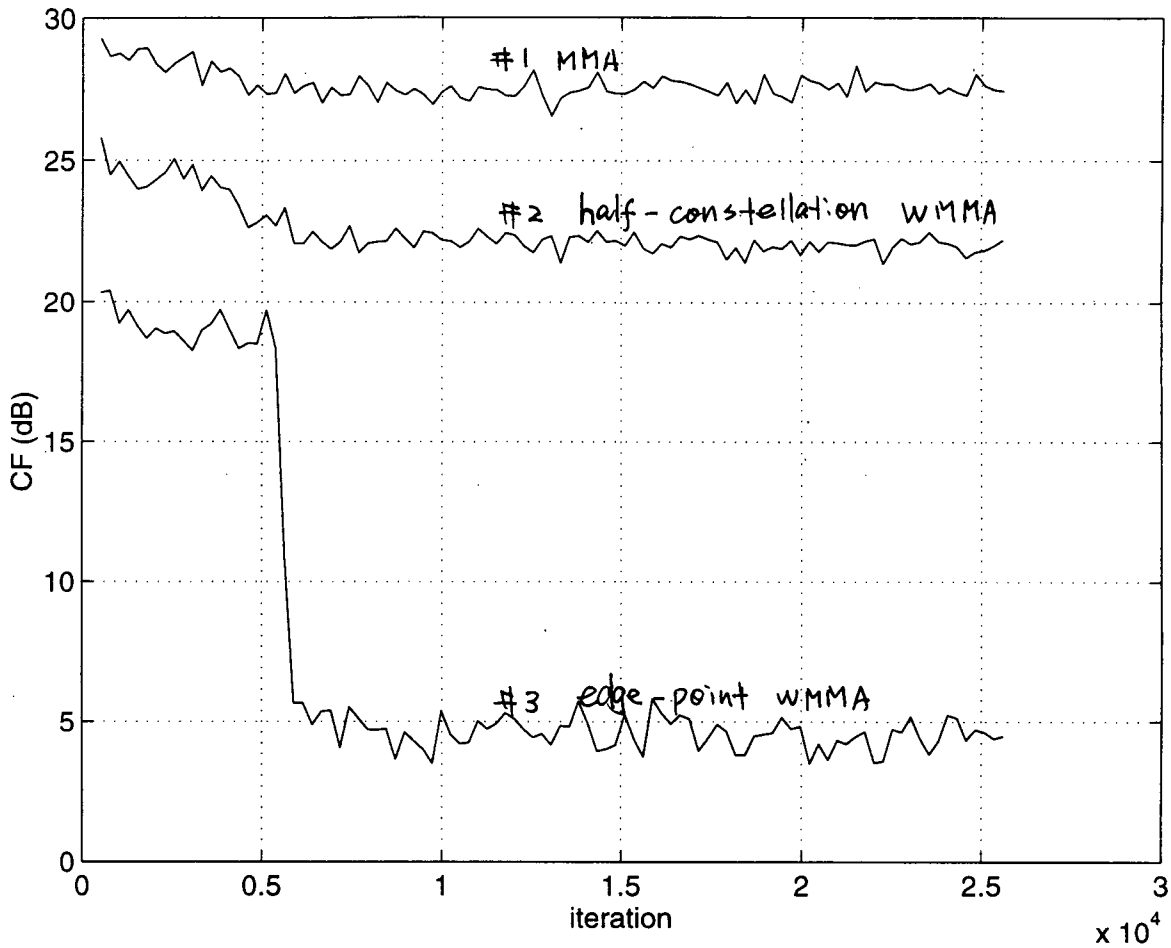


Figure 6.5: Cost function evolution for MMA and WMMA when used with 64-CAP

the agreement between the theoretical values calculated in Table 5.2 and the experimental results obtained in Fig. 6.5 is excellent.

Evolution of the MSE

The simulations demonstrate how well blind convergence is improved with various versions of WMMA. Fig. 6.6 shows the MSE comparison of MMA, the half-constellation WMMA, and the edge-point WMMA. The equalizer achieves the best steady-state blind convergence performance with edge-point WMMA.

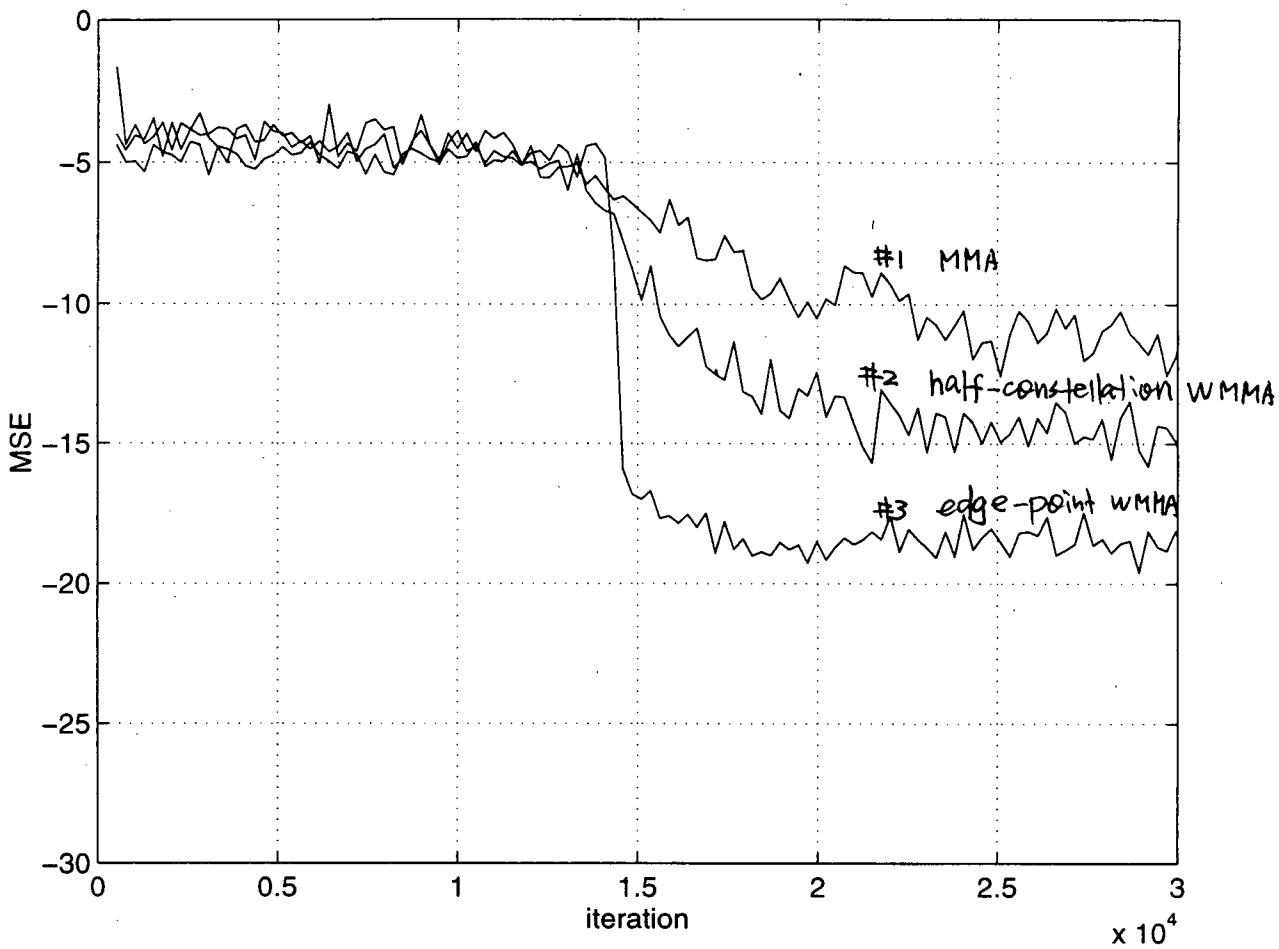


Figure 6.6: MSE learning curves for various MMAs

In Figs. 6.7 through Fig. 6.10, we show the plots of converged 64-point signal constellations for various algorithms. Fig. 6.7 shows the signal constellation after converging with MMA, and Fig. 6.10 shows the result with LMS. The two figures tell us that even though we can achieve initial convergence with the blind algorithm, we have to rely on the LMS algorithm to obtain the optimum steady-state performance. Fig. 6.8 shows the converged constellation with the half-constellation WMMA, and Fig. 6.9 shows the constellation obtained with the edge-point WMMA. The convergence performance is improved by using the half-constellation WMMA, and is further improved by using the edge-point WMMA. From Figs. 6.9 and Fig. 6.10, we see that almost the

same steady-state convergence performance is achieved by using the edge-point WMMA and the LMS algorithms. Note that the step size used for the edge-point WMMA is about five times larger than for the other algorithms.

The use of windowed MMA, especially edge-point WMMA, is limited to applications with a limited number of symbol levels. The number of edge points in a signal constellation is proportional to the number m of symbol levels, and the number of signal points in the constellation is proportional to m^2. If we assume that all the symbols have the same probability of occurence, then the probability of sending edge points is inversely proportional to m. Thus, for very large signal constellations, reasonable convergence speed cannot be obtained due to the lack of a sufficient number of iterations during filter adaptation.

6.3 Laboratory Experiments

Experimental results are not yet available for the 155 Mb/s 64-CAP ATM LAN application. However, the basic MMA blind equalization algorithm has recently been implemented in real time on two different laboratory experimental setups for two other applications. The first experimental setup implements the 51.84 Mb/s 16-CAP transceiver specified by DAVIC in [5] for FTTC networks. The second experimental setup implements several asymmetric digital subscriber line (ADSL) transceivers, which use a variety of signal constellations and provide data rates in the megabit per second range. In both cases, the MMA algorithm is performing very well, and has demonstrated its superiority over the RCA algorithm which was previously used. Experimental comparison with CMA has not been done yet.

The experimental results presented here are for the experimental 51.84 Mb/s 16-CAP transceiver specified by DAVIC. This transceiver has to operate in an environment that is much harsher than the ATM LAN environment, and the experimental results described here reflect this fact. Specifically, this application requires the usage of a decision feedback equalizer (DFE), which is more complex than the linear equalizer considered in the rest of this thesis, and is also significantly more difficult to train blindly. As a result, since they were available, we decided to also show some experimental results obtained with a DFE. This DFE uses MMA for the convergence of the feedforward filter,

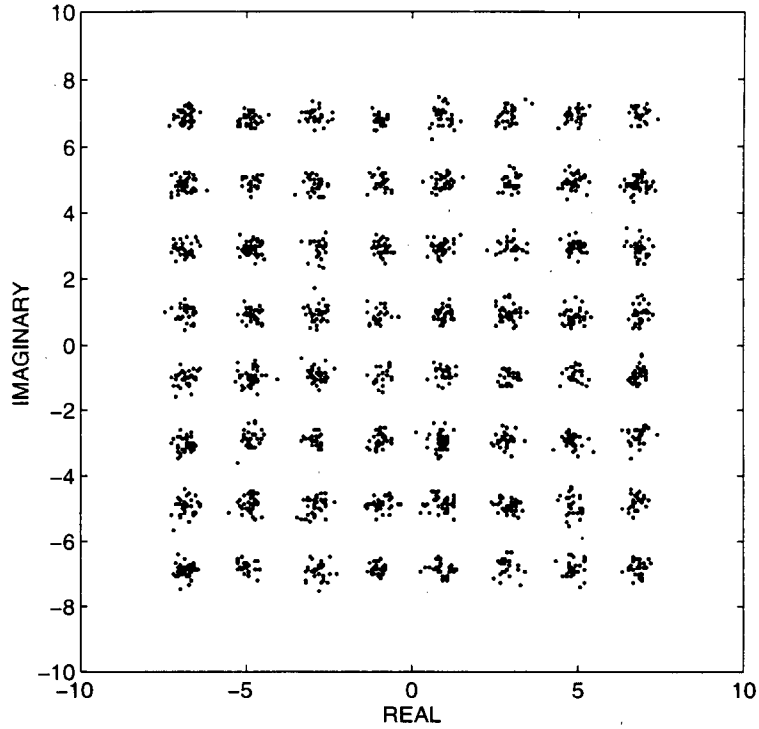


Figure 6.7: After convergence with MMA

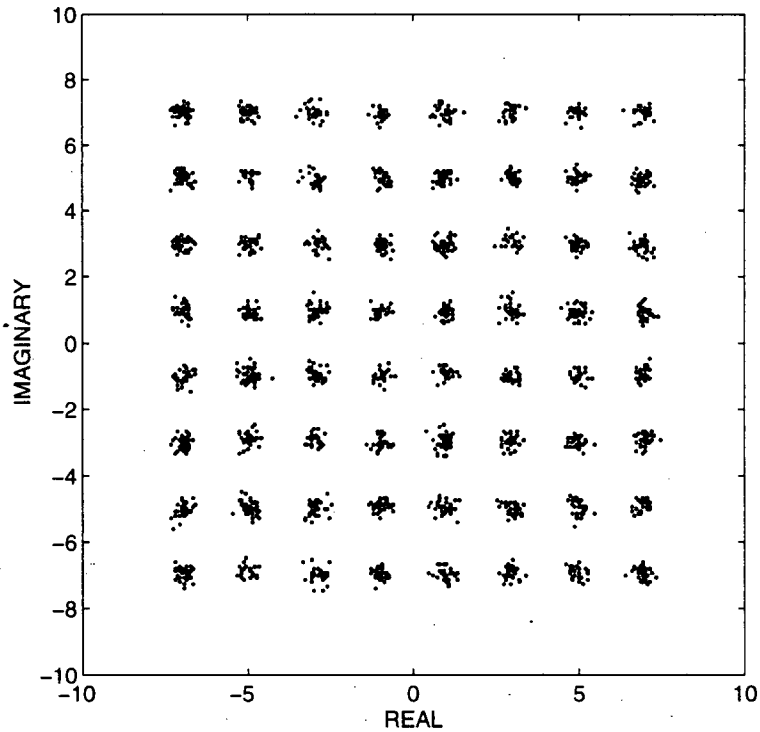


Figure 6.8: After convergence with half-constellation WMMA

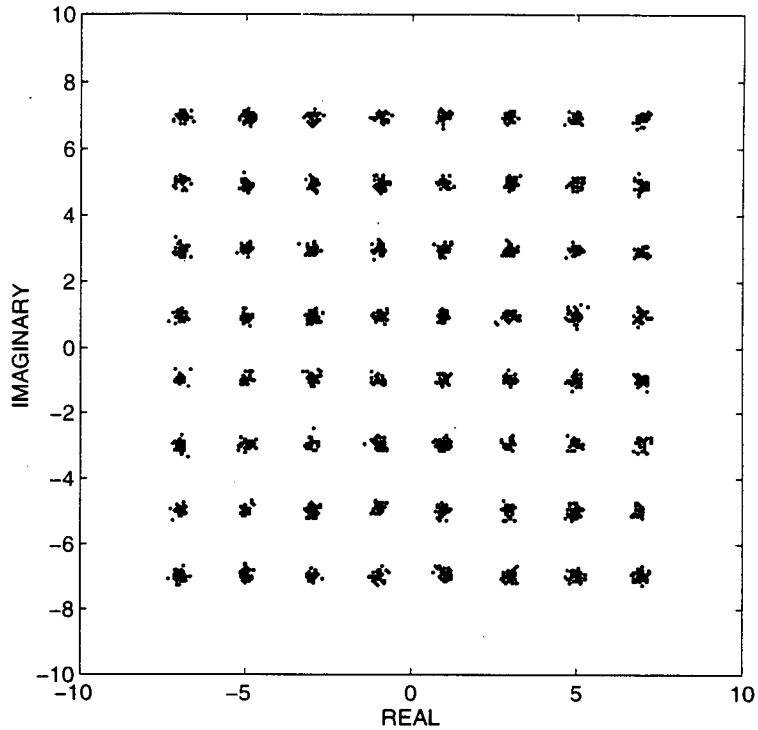


Figure 6.9: After convergence with edge-point WMMA

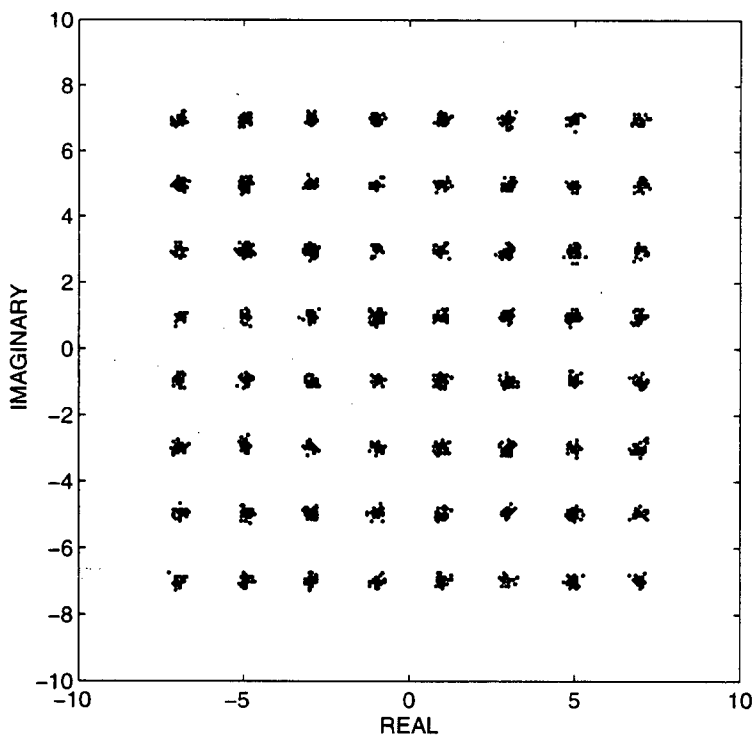


Figure 6.10: After convergence with LMS algorithm

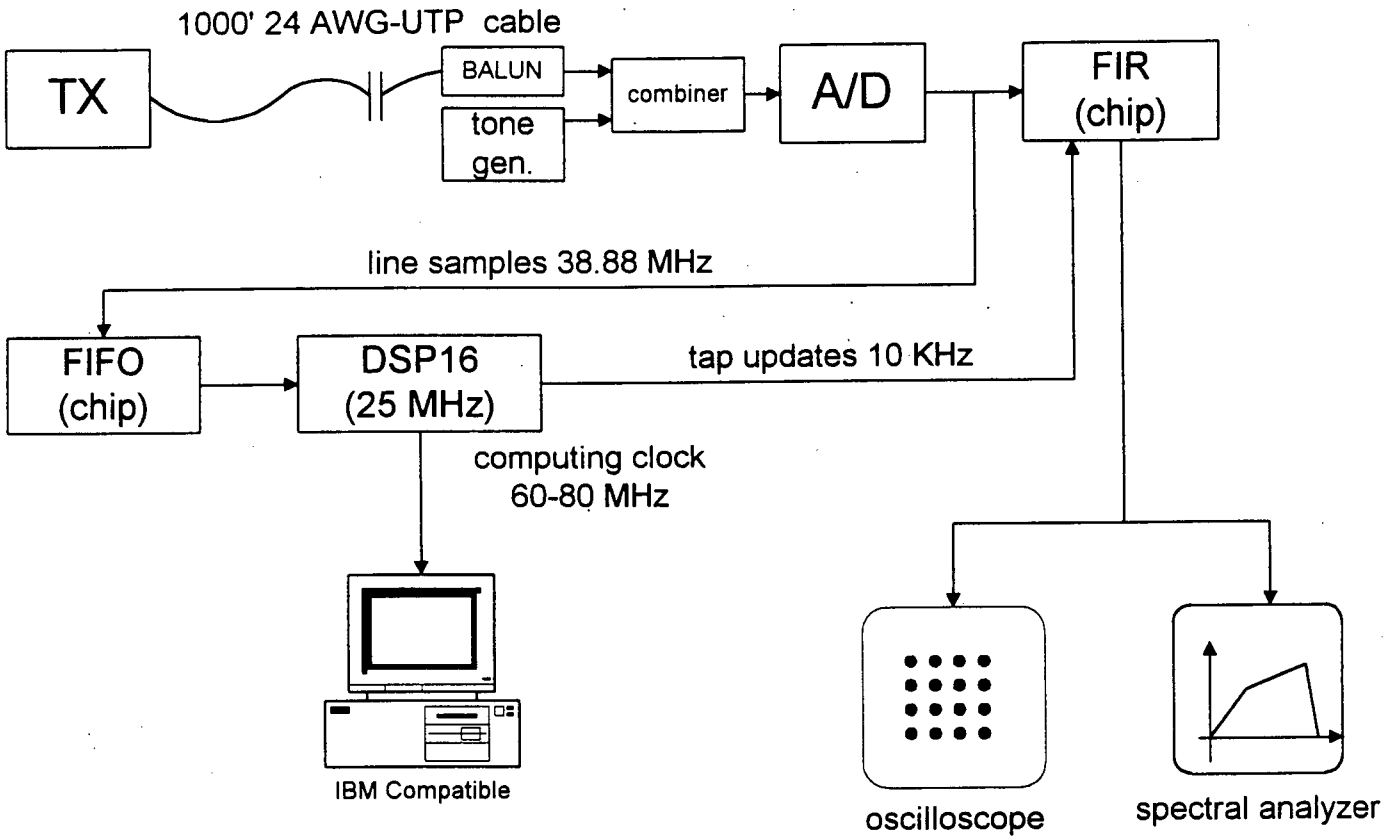


Figure 6.11: DSP16 setup in lab experiment

as described in [30]. The purpose for showing these results is that they highlight another major advantage of MMA over both RCA and CMA, which is the fact that MMA can readily be used with a DFE, whereas the two other algorithms cannot. In one of the experiments described later, MMA was able to open the eye for the DFE under extremely severe conditions of channel impairment and interference, whereas RCA was completely unable to do so.

6.3.1 Testing Environment

Fig. 6.11 shows the real-time and real world DSP setup used in the laboratory. In this setup, signals are transmitted over a 1000 foot 24 American wire gauge (AWG)-UTP cable. When a noisy environment is simulated for RF interference, a single tone generator is connected to the receiver through a balance/unbalance (BALUN) connector and a combiner. The receiver is partitioned into two main blocks. One consists of two very fast FIR filter chips, which implement a $T/3$ FSLE whose inputs are taken from the A/D at the sampling rate of 38.88 MHz, and the outputs are computed at the symbol rate of 12.96 MHz. The second block consists of a first-in-first-out (FIFO) chip and a DSP chip from Lucent Technologies called DSP16. These devices are used to implement the tap updating algorithms at a rate which is much lower than the symbol rate. The input of the FIFO chip is connected to the A/D which generates real world line samples at 38.88 MHz. The outputs of the FIFO chip are sent to the DSP16 board which is monitored by an IBM-compatible computer. The filter taps are updated in the DSP16 board which has a master clock of 60~80 MHz. The tap coefficients are then downloaded to the real-time FIR chips at a rate of 10 kHz. In this chip, the output of the equalizer, $y_n = \mathbf{c}_n^T \mathbf{r}_n$, is computed at the symbol rate of 12.96 MHz for 16-CAP. These outputs of the FIR filter are sent to an oscilloscope and a spectral analyzer to display the signal constellation and frequency response of the equalizer.

6.3.2 Experimental Results

In these experiments, we have tested MMA in several environments: MMA with no additive noise, MMA with a single tone RF interferer and MMA with noise and with DFE.

First,we tested MMA with no additive noise. Figs. 6.12 (a) through (e) show the experimental results obtained with this test. The initial signal constellation without equalization is shown in Fig. 6.12 (a). Figs. 6.12 (b) through Fig. 6.12 (d) show the constellations at 10 sec. intervals during blind convergence. Fig. 6.12 (e) gives the steady-state convergence with the LMS algorithm. Note that we use 10 secend interval in order to take photos of convergence. Experiments show that RCA has a similar performance to that of MMA, and these results are not shown here. It should be pointed out that speed of convergence was slowed down on purpose, so that good pictures could be

taken. The targeted convergence time for products is less than one second.

Then we tested the MMA blind algorithm with noise. A single tone RF interferer is injected into the channel as shown in Fig. 6.11. Fig. 6.13 displays the spectra of the signal and this interference, when SNR≈0 dB and the noise is centered at 5 MHz.

We tested two linear blind equalization algorithms with the single tone RF interferer. Fig. 6.14 (a) shows the signal constellation with MMA at 10 sec. intervals and Fig. 6.14 (b) shows the steady-state performance. Fig. 6.14 (c) through Fig. 6.14 (e) show the convergence with RCA at 10 sec. intervals. Comparing the two groups of tests with MMA and RCA, we can see that MMA has a faster convergence rate and has better convergence performance. It shows that MMA can achieve more reliable convergence than RCA in the presence of noise. However, these results also demonstrate that an FSLE is not well equipped to handle RF interference as can be seen from the bad steady-state eye openings in Fig. 6.14. This explains the need to use a decision feedback equalizer (DFE) consisting of a feedforward linear equalizer and a feedback filter. The details of operation of the DFE are given in [30]. The tap updating algorithm uses MMA for the feedforward filter and a 16-point slicer for the feedback filter. We show the blind convergence performance in Fig. 6.15 (a) through Fig. 6.15 (b). Fig. 6.15 (c) shows the steady-state convergence when the equalizer uses the LMS algorithm and the DFE. Comparing Figs. 6.12 (e) and Fig. 6.15 (c), we can see that almost the same convergence performance is achieved, where the former figure shows MMA with no noise and the latter one shows MMA with DFE in the presence of a single tone RF interferer. No results are shown for RCA, because RCA is not able to converge when the DFE is used.

6.4 Remarks

Various MMA algorithms have been tested for several applications. We conclude from these numerous tests that MMA and its generalized algorithms can achieve improved blind convergence.

We first made a convergence comparison between RCA and MMA. For the ATM LAN application, the equalizer converges faster with MMA than with RCA. MMA is very well suited to difficult applications, such as applications using high-order constellations. In contrast, the use of the four-

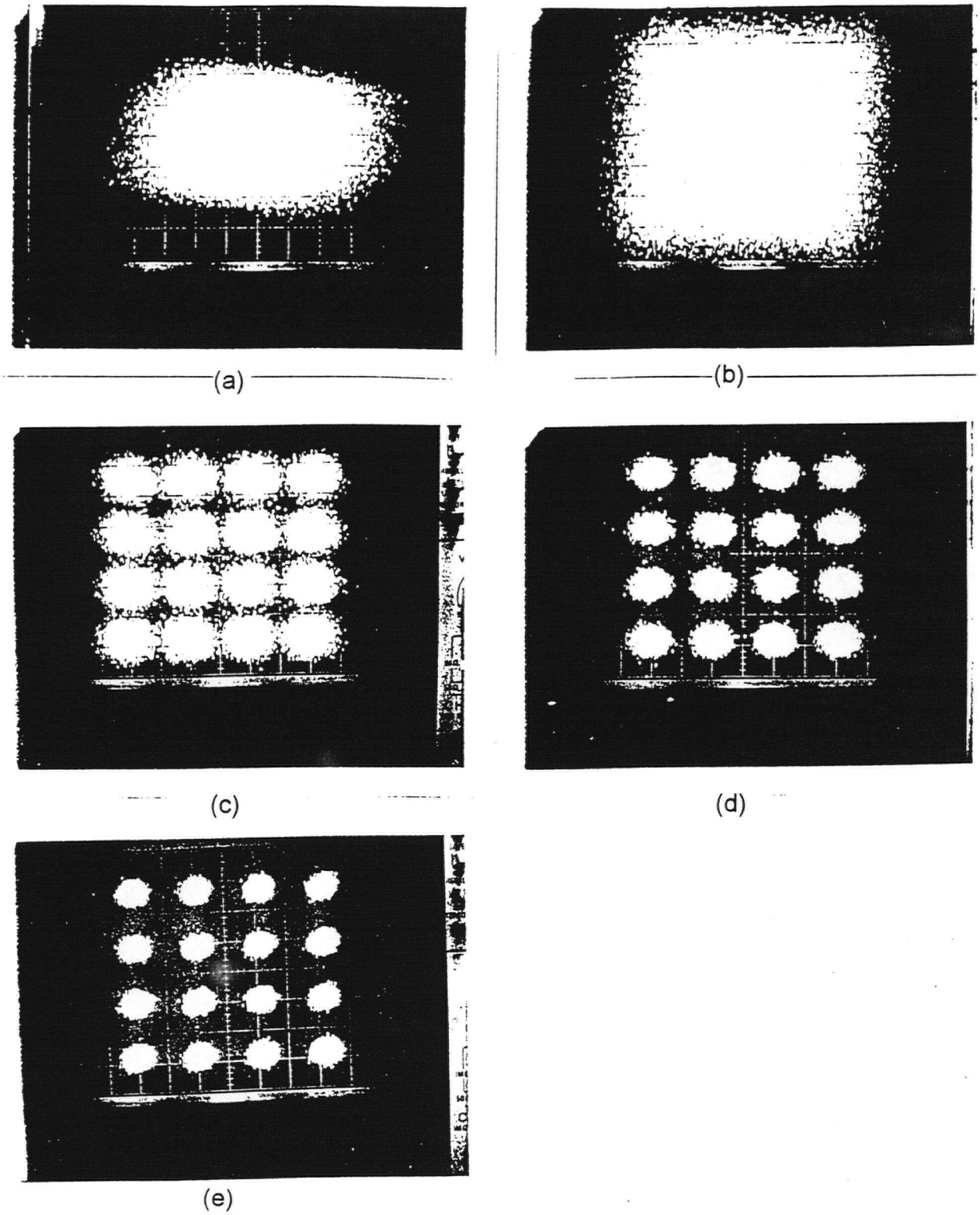


Figure 6.12: DSP test: MMA without noise

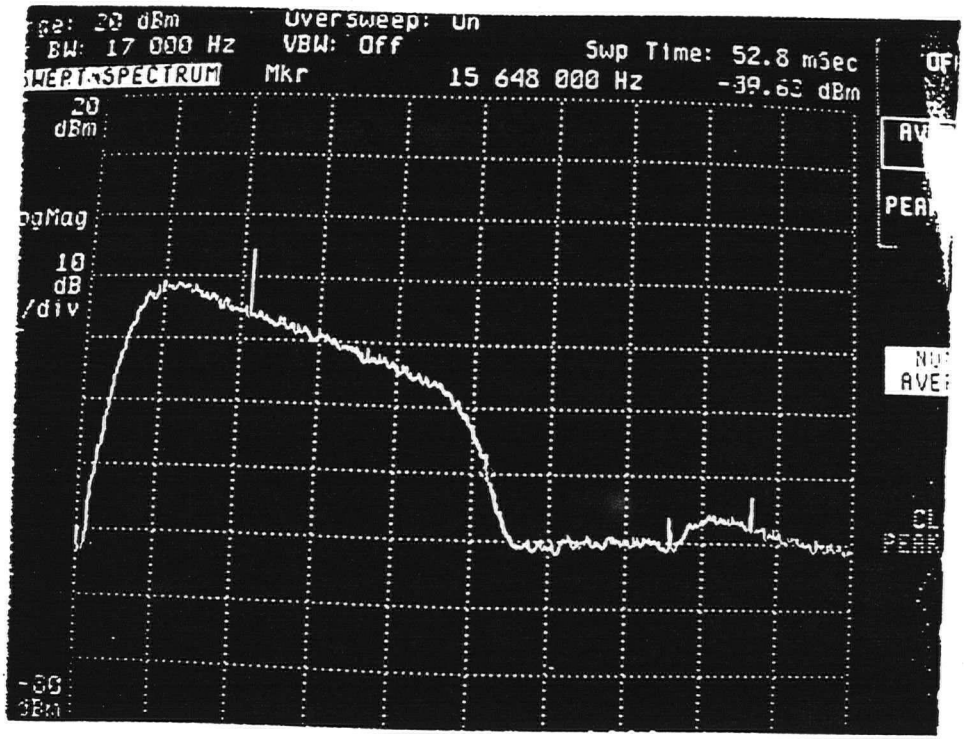


Figure 6.13: Transmitted signal with RF interference

point constellation limits the use of RCA to simple applications. MMA's usefulness has been demonstrated for the 64-CAP ATM LAN, 16-CAP FTTC and C-CAP ADSL applications, and will be incorporated in future systems for these applications.

Table 6.2 gives our recommendations for the type of blind equalization algorithms that should be used for various CAP applications.

Two more comments would be pointed out as following. One comment is about applications. In this thesis, the use of MMA is limited to CAP applications, however, the algorithm is also appliable to QAM applications in terms of blind start-up. Another comment is given regarding to ISI. We

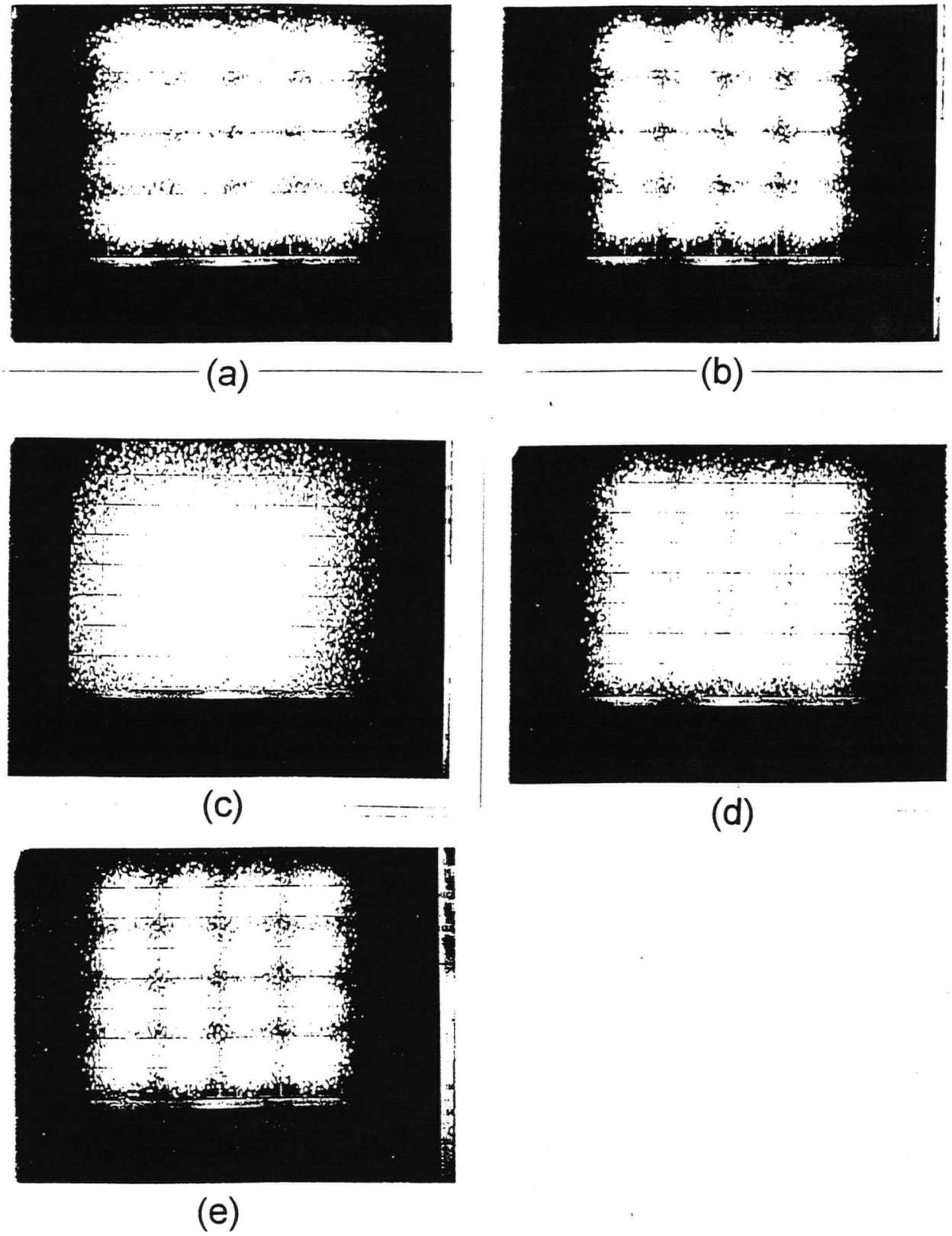


Figure 6.14: DSP test: MMA and RCA with RF interference and no DFE

4-CAP	16-CAP	32-CAP	64-CAP	128-CAP	256-CAP	512-CAP
RCA	MMA	MMA	MMA/WMMA*	MMA	GMMA	GMMA

*WMMA should be used if speed of convergence has to be improved.

Table 6.2: Choices of blind equalization algorithms

have proved ISI free for MMA in Chapter 3. GMMA and WMMA also have ISI free property and the proofs are not included in this thesis.

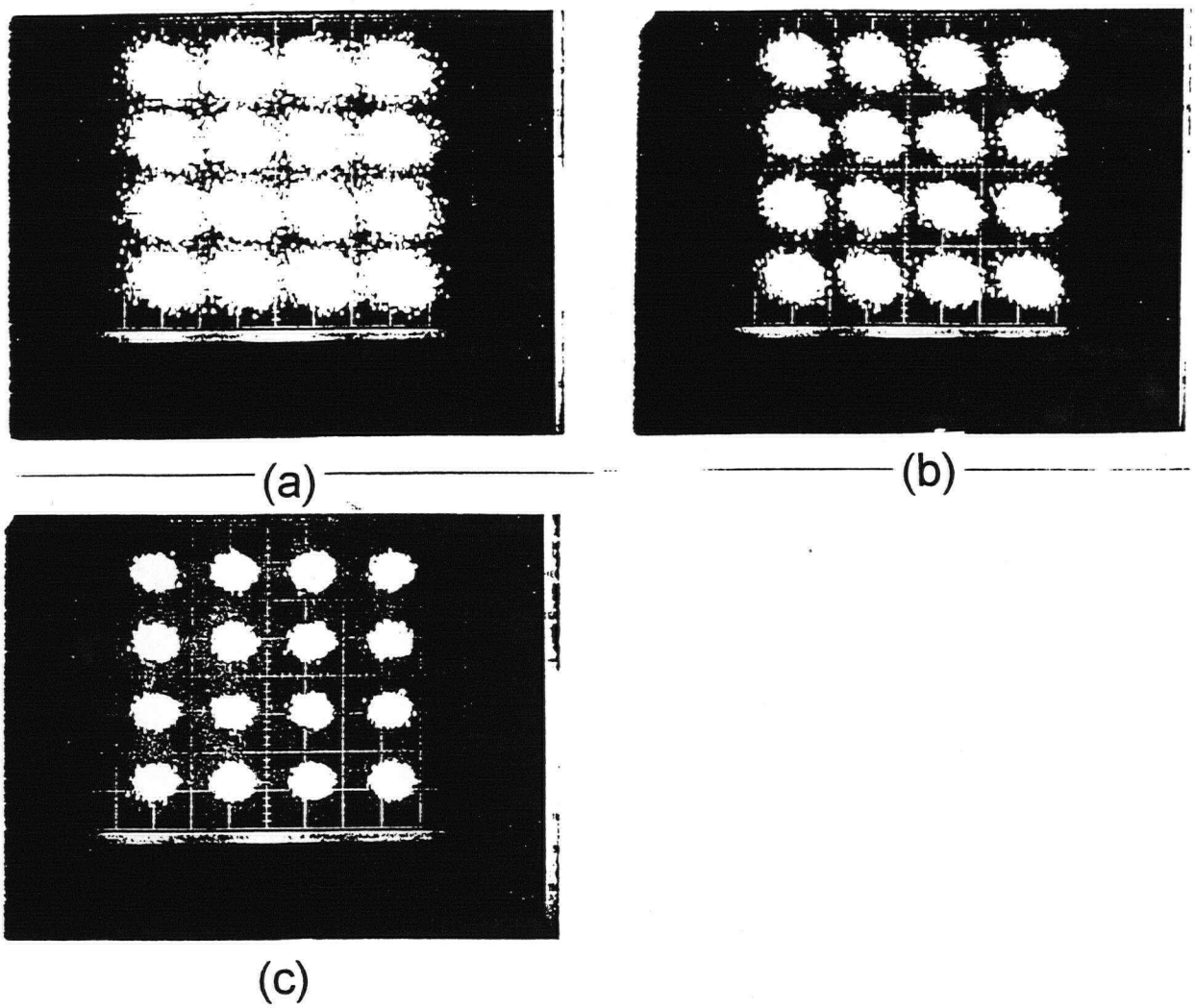


Figure 6.15: DSP test: MMA with RF interference and DFE

Chapter 7

Conclusions

In this thesis, we have proposed an improved blind algorithm, called MMA, and several other algorithms which are extensions of MMA. The contents of Chapter 3 through Chapter 5 give a detailed study of these algorithms, and Chapter 6 provides computer simulation and experimental results which confirm the theoretical results obtained in the previous chapters. In this chapter, we summarize the contributions, present the existing problems, and suggest further work.

7.1 Contributions

Several blind algorithms have been studied. The most commonly used are RCA and CMA. However, both algorithms cause some problems. RCA is simple to implement, but it often converges to wrong solutions, whereas CMA is immune to most wrong solutions but is expensive because it requires the addition of a rotator in the steady-state mode of operation. Based on these considerations, we have proposed a new algorithm called MMA to achieve better convergence reliability than RCA, and less complexity than CMA. The main feature of MMA is the use of multiple moduli which makes it much more flexible than RCA and CMA to accommodate a variety of different signal constellations. In this thesis, we have used multiple moduli for two different reasons. First, several moduli are required for nonsquare constellations, for which different subsets of symbols have different statistics. Multiple moduli are also required for applications with high-order constellations when one constant is not sufficient to represent the statistics of the symbols in a meaningful way. In summary, the use

of multiple moduli provides a means to solve problems which could not be handled by traditional blind algorithms. We summarize the contributions of the thesis as follows.

1. An improved blind equalization algorithm, called MMA, has been proposed [29] [30]. For most standard blind algorithms, such as RCA and CMA, only one modulus is used during blind convergence. MMA makes use of several moduli which allows it to deal with more complicated applications than is feasible with standard blind algorithms. The use of multiple moduli is more effective for nonsquare constellations, when the transmitted symbols have different distributions, and leads to a reduction of wrong solutions, such as the 144-point solution for the 128-CAP application [30].

   Even for square constellations, the use of MMA has some advantages. In comparison to RCA, MMA is more reliable and provides a better speed of convergence because it uses higher-order statistics. Even though the implementation of the algorithm is more complicated than that of RCA, the cost can be reduced if a power-of-two multiplier can be used [6]. Also, MMA is more powerful than CMA, because it can compensate for linear phase, which cannot be done with CMA. Although MMA provides more reliability for initial convergence than RCA, the probability of converging to wrong solutions still exists, This can be improved by using other techniques which are less expensive to implement than the rotator required by CMA [30].

2. The usage of MMA requires the calculation of a constant R, which is a statistical function of the symbols a_n. R can be directly computed from the symbols a_n, but this approach is tedious. In this thesis, we have derived a closed-form expression for the constant R as a function of the number m of symbol levels. This makes the calculation of R very simple. Details on the computation of symbol moments and the constants R used for RCA, CMA, and MMA are provided in Appendix B. These closed-form expressions for R make the analytical investigations of cost functions easier.

3. An analysis of the cost functions used for blind algorithms has been presented. There is a major difference between the behavior of the LMS algorithm and the blind algorithms discussed here. The error used in the LMS tap updating algorithm essentially goes to zero for

a long-enough equalizer and when there is little noise introduced in the channel. As a result, the cost function minimized by the LMS algorithm, i.e. the MSE, also goes to zero. With blind equalization, the error used by the tap updating algorithm and the minimum of the corresponding cost function do both not go to zero. The result is that the values of the tap coefficients keep fluctuating even in steady-state operation. These tap fluctuations introduce random noise at the output of the equalizer and affect the quality of the eye opening.

Our research has shown that the quality of the eye opening achievable with blind equalization is a direct function of the minimum value of the corresponding cost function. That is, small values of this minimum result in good eye openings and large values results in bad eye openings. It has also been found that the minimum value of the cost function is an increasing function of the number of symbol levels used in the signal constellation, and this holds for all the blind equalization algorithms discussed here. The preceding results led to the design of new, enhanced MMA cost functions having small minimum values even for a large number of symbol levels.

4. The cost function analysis led us to the proposal of GMMA as a means to improve the equalizer's blind convergence performance for applications using a high-order signal constellation. This is done by using several constants R in the cost function rather than just one. The dispersion of subsets of output samples of the equalizer around each of the moduli R is then minimized. By properly choosing the various moduli it is possible to significantly reduce the minimum of the cost function, even for high-order signal constellation. An iterative algorithm, which uses an equal-energy type of criterion, has been proposed to compute the optimum values for the various constants R. The convergence performance of GMMA has been tested for the 256-CAP application, and the computer simulation results have demonstrated the validity of the concept.

5. The last enhancements of MMA proposed in this thesis are various forms of windowed algorithms called WMMA. These algorithms also use cost functions which have reduced minimum values. WMMA only uses one constant R in the cost function, as is done with standard MMA,

but the algorithm does not use all the output samples of the equalizer. In one implementation of WMMA, only the outer half of the signal constellation is utilized. In an alternative approach, only the edge points of the signal constellation are utilized. Both approaches provide convergence improvements over the standard MMA for the 64-CAP application. For denser constellations, WMMA, especially edge-point WMMA, suffers from the fact that only infrequent tap updates are performed, and convergence time can be unacceptably long.

7.2 Future Work

Blind equalization algorithms face two major problems, which are:

a) to provide a good initial eye opening so that the receiver can switch reliably to the LMS algorithm, and

b) to avoid converging the equalizer to wrong solutions.

1. This thesis has mostly concentrated on issue **a)** above, and several new blind equalization algorithms, which achieve the desired goal for signal constellations with up to 256 points, have been presented, analyzed, and tested through computer simulations. However, even GMMA, the most powerful algorithm, has not been very effective with transceivers using signal constellations with more than 512 points. Investigating cost functions which can be used for these very dense signal constellations is one possible topic of future work,

2. Issue **b)** above has only been briefly mentioned in this thesis, but has also been the subject of intense research, which is documented elsewhere [30]. MMA is more reliable than RCA, but occasionally it can still converge the equalizer to wrong solutions. Computer simulations have shown that the technique proposed in [30] solves this problem. However, no analytical investigations have been conducted so far to provide some theoretical foundations for the techniques. This could be another topic for future work.

3. The relationship between the steady-state MMA cost function and the MSE has been discussed in Appendix C. Some simplifying assumptions were made in the analysis presented

there. One possible research topic is a more general analysis which does not use some of the assumptions.

4. Our study of blind equalization has shown that many factors influence the initial speed of convergence of the equalizer. Typical examples are: type of blind algorithms; order of the statistics used by the algorithm, equalizer structures; equalizer design parameters, such as step size, number of taps, and type of signal constellation; type of channel impairments and additive interference; and so forth. Given all these variables, the study of the initial convergence of blind equalizers is believed to be a major undertaking, and it is not surprising that this topic has not been addressed yet in the literature. However, it could, potentially, be a very fruitful research topic.

5. The MMA algorithm has been implemented on two experimental setups in the lab. This will allow real-time testing of the algorithm under channel impairment and noise conditions which cannot be easily simulated on the computer. Some new research topics may evolve from these experiments.

Appendix

A: Shaping Filter

The ATM LAN 64-CAP standard specifies square-root raised cosine shaping filters for the transmitter [1]. The impulse response $g(t)$ of a square-root raised cosine baseband filter is given by

$$g(t) = \frac{\sin[\pi(1-\alpha)t'] + 4\alpha t' \cos[\pi(1+\alpha)t']}{\pi t'[1-(4\alpha t')^2]} \tag{0.1}$$

where $t' = t/T$. The corresponding transfer function is given by

$$G(f) = \begin{cases} T & 0 \leq |f| \leq \frac{1}{2T}(1-\alpha) \\ \frac{T}{\sqrt{2}}\sqrt{1-\sin\frac{\pi T}{\alpha}(f-\frac{1}{2T}} & \frac{1}{2T}(1-\alpha) \leq |f| \leq \frac{1}{2T}(1+\alpha) \end{cases}$$

where the excess bandwidth is $\alpha = 0.15$ and the symbol rate is $1/T = 25.92$ Mbaud. The sampled values of the impulse responses of the in-phase and quadrature passband shaping filters are given by

$$s(iT') = g(iT')\cos(2\pi f_c iT') \qquad \tilde{s}(iT') = g(iT')\sin(2\pi f_c iT') \tag{0.2}$$

where the center frequency is $f_c = 15$ MHz, and the sampling rate is $1/T' = 77.76$ MHz. The impulse responses for the two shaping filters are plotted in Fig. 1.9. Fig. 1.10(a) shows the filters' frequency response in baseband, and Fig. 1.10(b) gives the filters' frequency response in passband.

Appendix

B: Calculation of the Constant R

This appendix presents the derivation of closed-form expressions for the various constants R used in the RCA, CMA, and MMA algorithms. As will be shown, these expressions can be conveniently expressed as a function of the number m of symbol levels.

The general approach used to compute the constant R will be explained for the MMA algorithm.

Constant R

The generalized form of MMA uses the following cost function .

$$CF = E[(|y_n|^L - R^L)^2 + (|\tilde{y}_n|^L - R^L)^2]] \tag{0.3}$$

where y_n and $\tilde{y_n}$ represent the equalizer's output samples, and L is a positive integer. Assuming the same statistics for y_n and $\tilde{y}_n$, we have $E[|y_n|^L] = E[|\tilde{y}_n|^L]$. For two-dimensional CAP systems, a_n and b_n represent the transmitted symbols for the in-phase and quadrature phase channels, respectively. When an equalizer converges, $y_n \to a_n$ and $\tilde{y}_n \to b_n$, and the cost function becomes

$$CF = E[(|a_n|^L - R^L)^2 + (|b_n|^L - R^L)^2]] \tag{0.4}$$

Assuming the same statistics for a_n and b_n, we have $E[|a_n|^L] = E[|b_n|^L]$. In the following, only the analysis for the in-phase channel will be provided. The same analysis applies to the quadrature phase channel. For the in-phase dimension, the gradient of the cost function in (0.3) with respect to the real tap vector was previously given in (3.24), and is repeated here

$$\nabla_{\mathbf{c}} = 2LE[(|y_n|^L - R^L)|y_n|^{l-2} y_n \mathbf{r}_n] \tag{0.5}$$

The constant R can now be evaluated by assuming a perfect equalization , i.e., $y_n \to a_n$, and by setting the gradient $\nabla_{\mathbf{c}}$ to zero. Also, if we assume that different symbols are uncorrelated, we get $E[a_n \mathbf{r}_n] = \mathbf{k}E[a_n^2]$, where $\mathbf{k}$ is a fixed vector whose entries are a function of the channel. We then get from (0.5)

$$E[(|a_n|^L - R^L)|a_n|^{L-2} a_n \mathbf{r}_n] = 0 \tag{0.6}$$

$$\mathbf{k} * E[(|a_n|^L - R^L)|a_n|^{L-2} a_n^2] = 0 \tag{0.7}$$

$$E[(|a_n|^L|a_n|^{L-2}a_n^2 - R^L|a_n|^{L-2}a_n^2] = 0 \tag{0.8}$$

$$E[(|a_n|^L|a_n|^{L-2}a_n^2] - R^L E[|a_n|^{L-2}a_n^2] = 0 \tag{0.9}$$

Solving for R^L we get

$$R_{mma}^L = \frac{E[|a_n|^{2L-2}a_n^2]}{E[|a_n|^{L-2}a_n^2]} = \frac{E[a_n^{2L}]}{E[|a_n|^L]} \tag{0.10}$$

By using the same method, we obtain the following expression for the constant R^L used for generalized CMA

$$R_{cma}^L = \frac{E[|A_n|^{2L}]}{E[|A_n|^L]}. \tag{0.11}$$

and for RCA

$$R_{rca} = \frac{E[a_n^2]}{E[|a_n|]} \tag{0.12}$$

Note that RCA is a second-order statistics algorithm, and does not have a generalized form.

The constant R has been derived for three blind algorithms. In practice, $L = 2$ is commonly chosen, in which case the constants R are given by

$$R_{cma}^2 = \frac{E[|A_n|^4]}{E[|A_n|^2]} \tag{0.13}$$

$$R_{mma}^2 = \frac{E[a_n^4]}{E[a_n^2]} \tag{0.14}$$

$$R_{rca} = \frac{E[a_n^2]}{E[|a_n|]} \tag{0.15}$$

We see that the constants R are the statistical functions of the symbols A_n or a_n.

Square Constellations

The expressions derived for the constants R in the previous section are functions of the moments of the symbols A_n and a_n. These moments can be computed on an individual basis, although this could be tedious. For the usual case where the symbols have odd integer values it is possible to derive simple closed-form expressions for the constants as a function of the number of symbol levels. Assume that the symbols takes the following values used by C-CAP system,

$$\{a_n\} = \{\pm 1, \pm 3, \ldots, \pm(2m-1)\} \tag{0.16}$$

where m indicates the number of symbol levels. If C indicates the number of constellation points, m can be directly derived from

$$m = \frac{\sqrt{C}}{2} \tag{0.17}$$

The following summations can be found in [12]

$$\sum_{k=1}^{m} k = \frac{1}{2}m(m+1) \tag{0.18}$$

$$\sum_{k=1}^{m} k^2 = \frac{1}{6}m(m+1)(2m+1) \tag{0.19}$$

$$\sum_{k=1}^{m} k^3 = \frac{1}{4}[m(m+1)]^2 \tag{0.20}$$

$$\sum_{k=1}^{m} k^4 = \frac{1}{30}m(m+1)(2m+1)(3m^2+3m-1) \tag{0.21}$$

These summations do not apply directly to sums of powers of odd integer, but can be used to derive closed-form expressions for these types of summations. For example, we can write

$$\sum_{k=1}^{m}(2k-1) = \sum_{k=1}^{2m-1} k - 2\sum_{k=1}^{m-1} k = m^2$$

where the two sums in the middle have been evaluated from (0.18). Similar summation manipulations can be used for other sums of powers of odd integers, and we get

$$\sum |a_n| = \sum_{k=1}^{m}(2k-1) = m^2 \tag{0.22}$$

$$\sum a_n^2 = \sum_{k=1}^{m}(2k-1)^2 = \frac{m}{3}(4m^2-1) \tag{0.23}$$

$$\sum |a_n|^3 = \sum_{k=1}^{m^2}(2k-1)^3 = m(2m^2-1) \tag{0.24}$$

$$\sum a_n^4 = \sum_{k=1}^{m}(2k-1)^4 = \frac{m}{15}(4m^2-1)(12m^2-7) \tag{0.25}$$

For square constellations, the probability of occurence of each symbol level is the same, i.e., $\frac{1}{m}$, so that the moments of the symbol levels become

$$E[a_n^L] = \frac{1}{m}\sum |a_n^L| \tag{0.26}$$

For CMA, the constant R is a function of the moments of the complex symbols A_n. Assuming that the symbols a_n and b_n are uncorrelated, it is easily verified that

$$E[|A_n|^2] = 2E[a_n^2]; \quad E[|A_n|^4] = 2E[a_n^4] + 2[E[a_n^2]]^2 \tag{0.27}$$

Using the above results, the constants R for the three blind algorithms can be expressed in the following simple way as a function of m

$$\begin{aligned} R_{cma}^2 &= \frac{E[|A_n|^4]}{E[|A_n|^2]} = \frac{56m^2 - 26}{15} \\ R_{mma}^2 &= \frac{E[a_n^4]}{E[a_n^2]} = \frac{12m^2 - 7}{5} \\ R_{rca} &= \frac{E[a_n^2]}{E[|a_n|]} = \frac{4m^2 - 1}{3} \end{aligned}$$

The constants R can be easily calculated for any given number of symbol levels m.

Appendix

C: Cost Function and the Eye Opening

In this appendix we present a steady-state analysis of the effect of the minimum value of the MMA cost function on the quality of the eye opening or, equivalently, the MSE. The analysis will be made for the output of the in-phase filter $\mathbf{c}_n$ of the adaptive equalizer. In order to carry through the analysis, we will have to make some simplifying assumptions. First, we will assume that the receiver has knowledge of the optimum tap vector $\mathbf{c}_0$ which eliminates ISI. Thus, if $\mathbf{r}_n$ is the vector of A/D samples in the tapped delay line in symbol period nT, the receiver can compute

$$\mathbf{c}_0^T \mathbf{r}_n = a_n \tag{0.28}$$

where a_n is the (ISI-free) symbol received in symbol period nT. In addition, we assume that the receiver uses the result in (0.28) to compute the correction term for the MMA tap updating algorithm and then adds this term to the optimum tap vector $\mathbf{c}_0$, so that

$$\mathbf{c}_{n+1} = \mathbf{c}_0 + \mu(a_n^2 - R^2)a_n\mathbf{r}_n \tag{0.29}$$

The output of the in-phase filter in symbol period $(n+1)T$ is then computed as

$$\begin{aligned} y_{n+1} &= \mathbf{c}_{n+1}^T \mathbf{r}_{n+1} && (0.30)\\ &= \mathbf{c}_0^T \mathbf{r}_{n+1} + \mu(a_n^2 - R^2)a_n\mathbf{r}_n^T\mathbf{r}_{n+1} && (0.31)\\ &= a_{n+1} + \mu(a_n^2 - R^2)a_n\mathbf{r}_n^T\mathbf{r}_{n+1} && (0.32) \end{aligned}$$

The term on the right is a noisy term introduced by the tap updating algorithm in (0.29), and is called tap adaptation noise. The process is then repeated. That is, in symbol period $(n+1)T$ the receiver computes (0.28) and (0.29) and then uses (0.32) to compute the output sample y_{n+2} in symbol period $(n+2)T$.

The scenario considered here is somewhat optimistic, because it assumes that the correction term in the tap updating algorithm can be computed with the symbol a_n and can then be added to the optimum tap vector $\mathbf{c}_0$. In practice, the equalizer's output sample y_n and the previously updated tap vector $\mathbf{c}_n$ have to be utilized. This more general case does not seem to be amenable to a

meaningful analysis. However, the results obtained with the simplified analysis presented here are consistent with the computer simulation results. Using (0.32), the MSE in the steady-state mode of operation can be written as

$$MSE = E[(y_{n+1} - a_{n+1})^2] = \mu^2 E[(a_n^2 - R^2)^2 a_n^2 (\mathbf{r}_n^T \mathbf{r}_{n+1})^2] \tag{0.33}$$

We now assume that the terms in the dot product $\mathbf{r}_n^T \mathbf{r}_{n+1}$ which are correlated with the symbol a_n make a small contribution to the value of the dot product. This is generally true if the equalizer has a large enough number of taps. The MSE in (0.33) then becomes

$$MSE = \mu^2 E[(a_n^2 - R^2)^2 a_n^2] E[(\mathbf{r}_n^T \mathbf{r}_{n+1})^2] \tag{0.34}$$

where the term on the right is a channel-dependent constant. Since we have $2m-1 = M \geq |a_n| \geq 1$, it immediately follows that

$$M^2 E[(a_n^2 - R^2)^2] \geq E[(a_n^2 - R^2)^2 a_n^2] \geq E[(a_n^2 - R^2)^2] \tag{0.35}$$

Replacing (0.35) in (0.34) we get the following upper and lower bounds for the MSE:

$$\mu M^2 E[(a_n^2 - R^2)^2] E[(\mathbf{r}_n^T \mathbf{r}_{n+1})^2] \geq MSE \geq \mu^2 E[(a_n^2 - R^2)^2] E[(\mathbf{r}_n^T \mathbf{r}_{n+1})^2] \tag{0.36}$$

The following discussion will concentrate on the lower bound of the MSE. Minimization of the expression on the right in (0.36) can be done by decreasing either the step size μ or the quantity $E[(a_n^2 - R^2)^2]$, which is the minimum of the MMA cost function. The other term in (0.36) in the product is not dependent on the design parameters of the MMA algorithm, and thus cannot be manipulated to minimize the MSE. This term, which is the dot product of vectors of A/D samples, can be considered to be a constant, because the receiver uses an AGC. This is not the case for the other terms.

An expression for the value of the minimum of the MMA cost function is given in (4.14) which is repeated here

$$E[(a_n^2 - R^2)^2] = \frac{16}{75}(12m^4 - 19m^2 + 7) \tag{0.37}$$

where m is the number of levels (in magnitude) of the symbols a_n. This is obviously an increasing function of m. Thus, if m increases the lower bound for the MSE in (0.36) also increases if the step

size μ stays constant. One way of keeping the bound constant would be to make μ variable and equal to

$$\mu^2 = \beta E^{-1}[(a_n^2 - R^2)^2] \tag{0.38}$$

where β is a constant. Such a choice is not desirable or feasible for large m, as shown in Table 0.1, which gives the value of the step size which keeps the MSE constant. First notice that the

C-CAP	4-CAP	16-CAP	64-CAP	256-CAP	1024-CAP
m	1	2	4	8	16
$E[(a_n^2 - R^2)^2]$	0	26.24	592	10,228	166,736
μ	-	3.0×10^{-2}	1.7×10^{-3}	9.8×10^{-5}	6.0×10^{-6}
μ^*	-	2^{-5}	2^{-9}	2^{-13}	2^{-17}

Table 0.1: Step sizes for C-CAP

minimum of the MMA cost function takes very large values when m becomes large. Notice also that there is a difference of almost four orders of magnitude between the step sizes required for 16-CAP and 1024-CAP if one wants to keep the bound on the MSE constant. Speed of convergence of the equalizer is (roughly) proportional to the value of the step size. Thus, if it takes about one second to converge with 16-CAP, it would take minutes to achieve the same with 1024-CAP, which is unacceptable.

The last row in the table gives the power of two μ^* which is the closest to the computed value of μ. Notice that this quantity decreases from 2^{-5} for 16-CAP to 2^{-17} for 1024-CAP. In practice, this means that the digital precision required for the 1024-CAP tap updating algorithm would be about 12 bits larger than what is required for 16-CAP. This, again, would not be acceptable for many applications.

Bibliography

[1] "The ATM 155.52 Mb/s Physical Layer Specification for Category-3 Unshielded Twisted Pair", *ATM Forum Spefication*, 1995

[2] Benveniste, A., and Goursat, M., "Blind Equalizers", *IEEE Trans. Commun.*, Vol. 32. No. 8, pp. 871-883, 1984

[3] Benvenuto, N., and Goeddel, T. W. "Classification of Voiceband Data Signal Using the Constellation Magnitude", *IEEE Trans. Commun.*, Vol. 43. No. 11, pp. 2759-2770, 1995

[4] Chen, W. Y., Im G. H. and Werner J. J., "Design of Digital Carrierless AM/PM Transceiver", *ANSI Contribution T1E1.4/92-149*, Aug. 1995

[5] "Lower Layer Protocol and Physical Interferace", *DAVIC 1.0 Specification Part 08*

[6] Duttweiler, D. L., "Adaptive Filter Performance with Nonlinear in the Correlation Multiplier", *IEEE Trans. Acoustic, Speech and Signal Processing*, Vol. ASSP. 30, No. 4, 1982

[7] Foschini, G. J., "Equalizing without Altering or Detecting Data", *AT&T Tech. J.*, Vol. 64, No. 8, pp. 1885-1911, Oct. 1985

[8] Gitlin, R. D., and Werner, J. J., "Blind Equalizer Tap-Adjustment Algorithm for Quadrature Amplitude Modulation Data Transmission", *Bell Laboratories Engineer's Notes*, Oct. 19, 1979

[9] Gitlin R. D., Hayes, J. F., and Weinstein, S. B., *Data Communications Principles*, Plenum Press, 1992

[10] Godard, D. N., "Method and Device for Training an Adaptive Equalizer by Means of an Unknown Data Signal in a QAM Transmission System", *U.S. Patent* 4 227 152, Oct. 7, 1980

[11] Godard, D. N., "Self-Recovering Equalization and Carrier Tracking in Two-Dimesional Data Communication System", *IEEE Trans. Commun.*, Vol. 28, No. 11, pp. 1867-1875, Nov, 1980

[12] Gradshteyn, I. S. and Ryzhik, I. M. *Table of Integrals, Series, and Products*, Academic Press, 1994

[13] Harman, D. D., Huang, G., Nguyen, M-H., Werner, J. J., and Wong, M. K., "Local Distribution CAP for IMTV", *IEEE Multimedia*, Vol. 2, No. 3, pp. 14-23, 1995

[14] Haykin, S., *Communication Systems*, John Wiley & Sons,1994

[15] Haykin, S., *Adaptive Filter Theory*, Prentice Hall, 1996

[16] Haykin, S., *Blind Equalization*, Prentice Hall, 1996

[17] Im, G. H., Harman D. D., Huang, G, Mandzik, A. V., Nguyen, M.-H., and Werner, J. J., "51.84 Mb/s 16-CAP ATM LAN Standard", *IEEE J. Selected Areas Commun.*, Vol. 13, No.4, pp. 620-632, 1995

[18] Im, G. H. and Werner, J. J., "Bandwidth-Efficient Digital Transmission over Unshielded Twisted-Pair Wiring", *IEEE J. Selected Areas Commun.*, Vol. 13, No.9, pp. 1643-1655, 1995

[19] Jablon, N. K., "Joint Blind Equalization, Carrier Recovering, and Timing Recovery for High-Order QAM Signal Constellations", *IEEE Trans. Signal Processing*, Vol. 40, No. 6, pp. 1383-1398, 1992

[20] Kennedy, R. A. and Ding Z., "Blind Adaptive Equalizers for QAM Communication Systems Based on Convex Cost Functions", *Opt. Eng.* Vol. 31, pp. 1189-1199, June, 1992

[21] Lee, E. A. and Meserschmitt D. G., *Digital Communication*, Kluwer Academic Publishers, 1994

[22] Mueller, K. H. and Werner, J. J. "A Hardware Efficient Passband Equalizer Structure for Data Transmission", *IEEE J. Trans. Commun.* Vol. COM-30, pp. 538-541, 1982

[23] Proakis, J. G., *Communication Systems Engineering*, Prentice-Hall, 1994

[24] Sato, Y., "A Method of Self-Recovering Equalization for Multilevel Amplitude-Modulation Systems", *IEEE Trans. Commun.*, pp. 679-682, June 1975

[25] Shalvi, O., and Weinstein, E., "New Criteria for Blind Deconvolution of Nonminimum Phase Systems (Channels)", *IEEE Trans. on Information Theory*, Vol. 36, No. 2, Mar., 1990

[26] Tugnait, J. K., "Identification of Linear Stochastic Systems via Second and Fourth-order Cumulant Matching", *IEEE Trans. Inform. Theory*, Vol. IT-33, pp, 393-407, May 1987

[27] Werner, J. J.,*Tutorial on Carrierless AM/PM-Part II*, ANSI Contribution T1E1.4/93-058, 1993

[28] Widrow B. and Strearns S. D., *Adaptive Signal Processing.* Prentice-Hall Inc., 1991

[29] Yang, J., Werner, J. J. and Dumont G. A., "The Multimodulus Blind Equalization Algorithm", *Accepted for publication at DSP97*, Santorini, Greece, 1997

[30] Yang, J., Werner, J. J. , "Blind Equalization", *Technical Memorandum*, Bell Laboratories, May 1996

Dupl. paging error

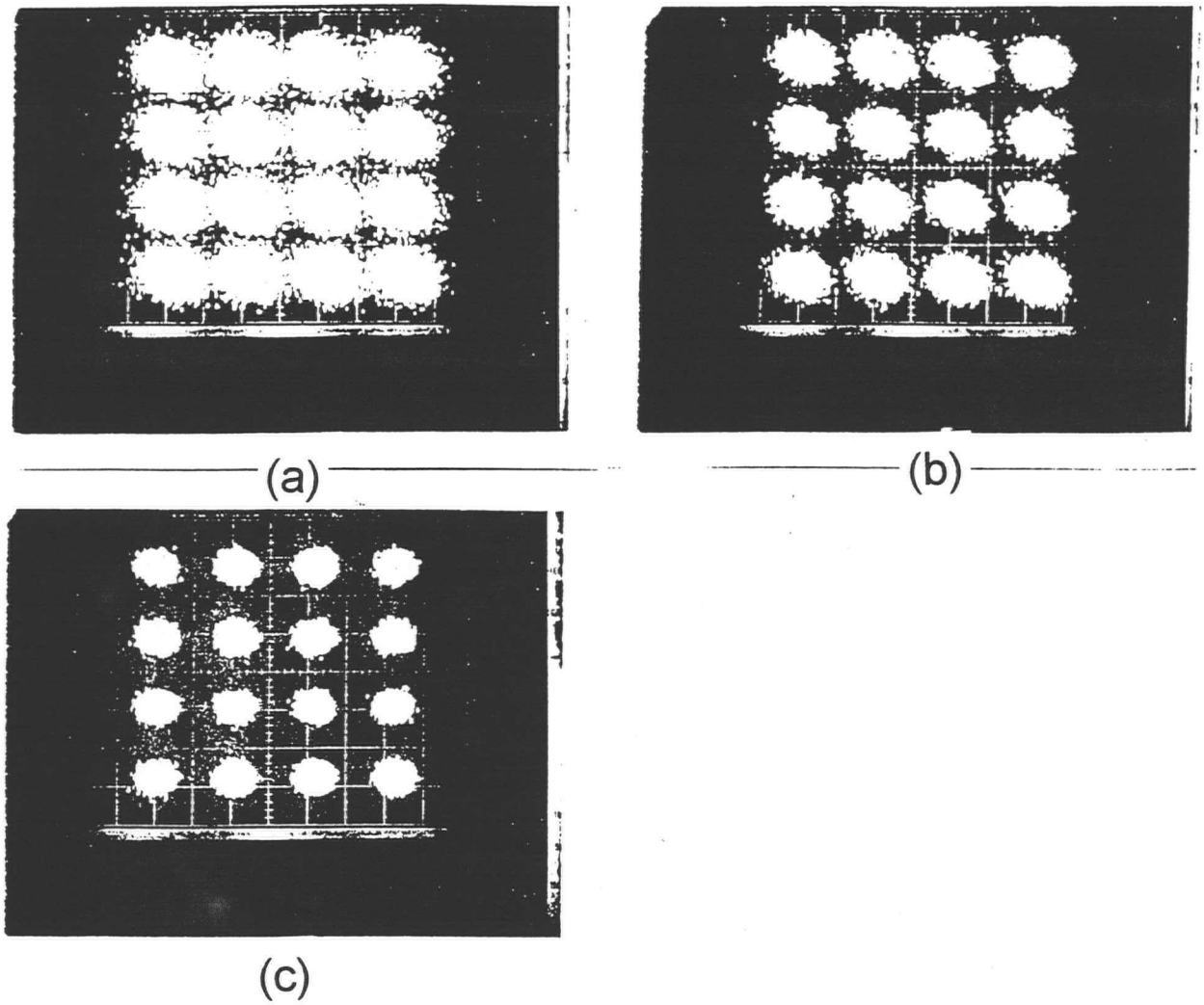


Figure 6.15: DSP test: MMA with RF interference and DFE