

THE FRACTIONALLY SPACED VECTOR
CONSTANT MODULUS ALGORITHM

BY

MARK ALDEN HAUN

B.S.E., Walla Walla College, College Place, 1996

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2002

Urbana, Illinois

© Copyright by Mark Alden Haun, 1999

ABSTRACT

The vector constant modulus algorithm (VCMA) was recently introduced as an extension of CMA which can equalize data from shaped sources having nearly Gaussian marginal distributions. VCMA was developed with the shaping technique of shell mapping in mind, but it also equalizes data from other shaping methods, like trellis shaping. Some simple changes in the structure of VCMA permit its use in fractionally-spaced equalizers, the benefits of which include shorter equalizer lengths, built-in matched filtering, and the possibility of global convergence. Although the global convergence of VCMA has not yet been proved, a novel approach to proving the convergence of CMA is presented, based on a view of the CMA cost function as a kurtosis-minimizing criterion; this provides some new insight into its behavior. VCMA can be seen as one member of a larger family of algorithms. The use of a vector modulus in the cost function has general applicability and is used to obtain the vector reduced constellation algorithm (VRCA) and the vector multimodulus algorithm (VMMA), two algorithms based on constant modulus criteria and derived from their scalar-modulus counterparts, RCA and MMA. These perform similarly to VCMA but have some unique characteristics. Simulations verify the superior performance of $T/2$ -spaced VCMA over baud-spaced VCMA on various short FIR channels, the ability of VCMA to equalize trellis-shaped data, and the performance of one of the related vector modulus algorithms.

ACKNOWLEDGMENTS

I wish to thank my advisor, Doug Jones, for his guidance, direction, and many helpful suggestions made during the course of my research on this thesis. I also wish to express my gratitude to my parents, whose love and encouragement have supported me throughout my education.

This research was supported by the Joint Services Electronics Program contract no. JSEP N00014-96-1-0129, and the Office of Naval Research contract no. N00014-95-1-0907.

TABLE OF CONTENTS

CHAPTER	PAGE
1 INTRODUCTION	1
1.1 Blind Equalization and the Constant Modulus Algorithm	1
1.2 Source Shaping	2
1.3 The Vector Constant Modulus Algorithm	4
2 SOURCE SHAPING METHODS	6
2.1 Shell Mapping	6
2.2 Trellis Shaping	7
3 THE FRACTIONALLY SPACED VECTOR CONSTANT MODULUS ALGORITHM	9
3.1 The Constant Modulus Algorithm (CMA)	9
3.2 Baud-Spaced VCMA	11
3.3 Fractionally Spaced VCMA	16
3.4 Normalized Fractionally Spaced VCMA	17
4 CONVERGENCE PROOFS BASED ON KURTOSIS CRITERIA	21
4.1 Kurtosis in the CMA Cost Function	23
4.2 Kurtosis of a Linear Combination of Random Variables	25
4.3 A Proof of Convergence for CMA	29
4.4 Constellations for which $E[X^2] \neq 0$	30
5 GENERALIZING THE VECTOR MODULUS CONCEPT TO OTHER ALGORITHMS	32
5.1 Vector Reduced Constellation Algorithm	33
5.2 Vector Multimodulus Algorithm	35

6	SIMULATIONS	37
6.1	Data Sets	37
6.2	Channels	39
6.3	FS-VCMA	40
6.3.1	Improvement of fractionally spaced over baud-spaced VCMA	40
6.3.2	Trellis-shaped data	42
6.3.3	Block size N	43
6.3.4	Block offset	45
6.4	FS-VRCA	45
7	CONCLUSIONS	47
	REFERENCES	49

LIST OF TABLES

Table	Page
3.1 Computational steps and their associated costs for one iteration of VCMA.	13
5.1 Computational steps and their associated costs for one iteration of VRCA.	34
6.1 Required number of iterations of FS-VCMA (in thousands) vs. block size N	44

LIST OF FIGURES

Figure	Page
3.1 Communications system model. A sequence of symbols $\{a_n\}$ from a set in the complex plane (16-QAM is shown here) are convolved first with the channel impulse response \mathbf{h} and then with the equalizer \mathbf{c} at the receiver.	10
4.1 Equalizer model showing the partition into a unit-energy vector \mathbf{c} and a scalar gain factor α	23
6.1 192-point constellation used for shell mapping.	38
6.2 256-point constellation used for trellis shaping.	39
6.3 Shell-mapped data after passage through channel \mathbf{h}_1	40
6.4 FS-VCMA equalizer output for input data in Figure 6.3.	41
6.5 FS-VCMA equalizer tap magnitudes vs. number of iterations ($\times 10^4$). . .	41
6.6 Baud-spaced VCMA equalizer output for input data in Figure 6.3.	42
6.7 Equalized trellis-shaped data through channel \mathbf{h}_1 using FS-VCMA.	43
6.8 Poor equalization of shell-mapped data with FS-VCMA when $N = 2$. . .	44
6.9 Equalized trellis-shaped data through channel \mathbf{h}_2 using FS-VRCA.	46
6.10 Equalized trellis-shaped data through channel \mathbf{h}_2 using FS-VCMA.	46

CHAPTER 1

INTRODUCTION

1.1 Blind Equalization and the Constant Modulus Algorithm

Any communication system may be decomposed into at least three parts: a sender or transmitter, a channel, and a receiver. Regardless of the type of signaling used, the channel usually introduces impairments of various kinds. In many applications, dispersion is the channel characteristic that most limits the system performance. For a digital communication system, this means intersymbol interference, or ISI: the tendency of the channel to smear together adjacent signaling pulses.

In many receivers, the task of “repairing” the channel-induced ISI on the received signal falls to the equalizer, a filter which, when cascaded with the channel response, will hopefully provide an overall channel free of dispersion. In many applications the equalizer takes the form of a finite impulse response (FIR) filter, a tapped delay line whose outputs are multiplied by the filter coefficients and then added together to obtain the output data. Since the channel may be time-varying and is seldom known ahead of time, an adaptive algorithm is typically used to adjust the filter coefficients based on a mutually agreed-upon training sequence in the transmitted data or perhaps symbol decisions fed back from the demodulator.

There are many situations where it is desirable to adapt an equalizer “blindly,” that is, without knowledge of the exact data being transmitted. For example, in many broadcasting situations, new receivers may come on-line at any time, and it would be inefficient to include frequent training signals in the transmitter’s data stream. Fortunately, side knowledge about the form and distribution of the transmitted signal is often enough to allow what is termed blind equalization.

The constant modulus algorithm (CMA), first described in [1] and [2], is a popular algorithm for blind equalization in digital communication systems. CMA uses a cost function which penalizes deviations of the received signal’s magnitude from a circle of constant modulus. Interestingly, CMA works well even with source signal constellations like QAM, in which the symbols are not of constant modulus. Its convergence ability does depend, however, on the probability distribution of the transmitted symbols. Specifically, the kurtosis of the transmitted data must be sub-Gaussian, a requirement that will be examined in further detail in Chapter 4. As the marginal source symbol distributions approach Gaussian, the convergence of CMA slows down and eventually fails altogether [3]. Unfortunately, this is at odds with source shaping techniques which attempt to increase system performance by using nonequiprobable signaling; an approximately Gaussian source is precisely the goal of source shaping.

1.2 Source Shaping

Source shaping is able to provide gain, independent of coding, by using efficient constellation shapes. Consider a simple example in two dimensions: A circular constellation has an inherent shaping gain of about 0.2 dB over a more traditional square containing the same number of signal points and unchanged spacing between the points, because the average energy of

all the constellation points is smaller. The symbol error probability will be similar for both constellations due to the common spacing, but the square constellation is wasting some energy on the corner points that would be better spread out evenly around the edges.

By choosing signal points from uniform spherical constellations in higher dimensions, higher gains are possible, approaching 1.53 dB for the shaping gain of an N -dimensional sphere over an N -dimensional cube as $N \rightarrow \infty$. The points chosen from the hypersphere are simply transmitted as a sequence of successive two-dimensional signal points. As the dimension of the underlying constellation increases beyond two, this has the effect of imposing a nonuniform probability distribution across the two-dimensional constellation. In the limit of large N , it is equivalent to imposing marginal Gaussian distributions on the two-dimensional constellation, which renders the signal unequalizable by the conventional constant modulus algorithm.

Although the gain achievable by shaping may appear insignificant, it is independent of the other coding schemes being used in a system. Thus, when most of the available computational resources of a system are needed to implement complex codes, the small incremental computation necessary to implement an algorithm like shell mapping becomes attractive. For this reason, source shaping is becoming widespread in applications; examples include the two most recent voiceband modem standards, V.34 and V.90, both of which use some form of shaping.

There are other types of modulation which produce near-Gaussian signals. Multicarrier modulation, such as OFDM (orthogonal frequency division multiplexing) and DMT (discrete multitone), inherently shapes the output data. These modulation types are finding widespread use in applications such as high definition television and other very high speed digital communications, because they allow a lower symbol rate and can reduce the complexity of the receiving hardware.

1.3 The Vector Constant Modulus Algorithm

As previously mentioned, CMA often fails to equalize shaped sources, as do other methods based on higher-order statistics (HOS). Algorithms based on second-order statistics (SOS) are expensive and have not yet been shown to work reliably [4]. Therefore, with shaped signal constellations becoming more common, an alternative blind equalization algorithm is clearly needed. The vector constant modulus algorithm (VCMA) [5] was recently introduced as an extension of CMA which can equalize data from shaped sources having nearly Gaussian marginal distributions. VCMA assumes that the source shaping is accomplished by choosing signal points uniformly distributed in a $2N$ -dimensional sphere and transmitting them as a sequence of N complex values. (In the case of shell mapping, discussed in the next chapter, this is in fact roughly how the shaping is done.) Vectors of N successive samples ought to be distributed uniformly in $2N$ -dimensional space, so a version of CMA which operates on *vectors* of the received samples should be able to equalize the shaped data. This is the key insight which explains how VCMA works.

This thesis presents some further results from the study of VCMA and related algorithms. Chapter 2 explores the applicability of VCMA to data which has been shaped by a method other than shell mapping. Although VCMA is a conceptual match to the shaping process used by shell mapping, other shaping methods have been described, and it is important to know whether or not VCMA works in these situations as well. Chapter 3 explains how VCMA may be modified for use in fractionally-spaced equalizers. This important result removes a barrier to the use of VCMA in many applications where fractionally-spaced equalizers are standard practice due to their ability to perfectly equalize many FIR channels and perform other receiver functions at

the same time. Also in this section, computational short-cuts are presented which reduce the cost of VCMA to the point at which it is competitive with ordinary CMA.

Chapter 4 considers a convergence proof for CMA based on a kurtosis criterion. This approach differs from that used by most other researchers and, although it is not directly applicable to VCMA, aids an intuitive understanding of CMA's convergence limitations.

It is natural to ask whether VCMA is part of a broader family of blind equalization algorithms using the vector modulus concept, much as CMA is part of a larger group of algorithms exploiting the predictable scalar modulus of transmitted data. Chapter 5 shows how some of these algorithms may be generalized to their vector modulus counterparts, thus making them also suitable for equalizing shaped modulation. Chapter 6 is devoted to simulations of the algorithms discussed in the previous chapters, and verifies the practicality of many of these variations on basic VCMA. This is followed with some conclusions in Chapter 7.

Throughout the text and formulas, vectors and matrices are represented by lower- and upper-case boldface letters, respectively. Nonconjugate transpose is indicated by a superscript letter T (T), and Hermitian (conjugate) transpose by a superscript letter H (H). Convolution is symbolized by an asterisk (*).

CHAPTER 2

SOURCE SHAPING METHODS

In theory, source shaping does not seem difficult to accomplish. One conceptually simple approach is to select points uniformly from a high-dimensional hypersphere and transmit them as a series of symbols from a two-dimensional base constellation. In practice, the challenge lies in finding a simple way to select only the points which are inside the hypersphere, while providing a way to index those points efficiently. A variety of techniques have been used for this task.

2.1 Shell Mapping

Shell mapping is the most common source shaping technique in current use and is the shaping method specified by the V.34 telephone modem standard [6]. In shell mapping, a two-dimensional circular constellation is divided into M rings of increasing energy. Each ring has an equal number of points and is assigned a cost label proportional to the energy of the points (often the labels are simply taken to be the integers counting rings outward). Now consider the $2N$ -dimensional region containing all of the points described by sequences of N symbols from the original two-dimensional constellation. Shell mapping provides an efficient way to select the 2^b lowest-cost signal points from this higher-dimensional set, where the cost is a measure of

signal energy determined by adding the N individual ring costs. The subconstellation selected by shell mapping in this way is very close to a sphere in $2N$ dimensions. Thus, b bits can be sent using N of the two-dimensional signal points transmitted in sequence. Inherent in this technique is the ability to send a fractional number of bits per symbol [7].

The applicability of VCMA to shell-mapped data is rather intuitive: By using vectors of the received signal points, the algorithm is able to operate on signal vectors having a uniform distribution and get around the problem of the almost-Gaussian marginal distributions of signal points in two dimensions [5]. By this reasoning, any shaping technique which achieves non-equiprobable signaling by projecting points from a higher-dimensional uniform constellation should produce data equalizable by VCMA.

2.2 Trellis Shaping

A different shaping approach, trellis shaping, was described by Forney in [8]. In contrast to shell mapping, which chooses points in a spherical constellation of finite dimension, trellis mapping searches the trellis of a convolutional code to find the lowest-energy sequence from a larger set of possible transmitted sequences.

A simple form of trellis shaping, which Forney calls “sign-bit shaping,” is accomplished by starting with a square two-dimensional QAM constellation. This constellation is partitioned into its four quadrants, with any given point being addressed by a combination of two sign bits to select the quadrant plus other bits that select the signal point within the quadrant. The original data sequence consisting of points in the square constellation is shaped by adding (modulo-2) its sequence of sign bits to a sequence in the convolutional code, thereby scrambling the quadrants of the original points in a constrained manner. A Viterbi algorithm search

through the trellis of the convolutional code is performed using the energies of the resultant modified signal points as the branch metrics in order to select the convolutional codeword that minimizes the transmitted signal energy.

It is not possible to describe this operation in a specific, finite-dimensional space, yet simulations (presented later in this paper) have shown that VCMA is able to equalize trellis-shaped data. It is hypothesized that the required data-vector-length N in VCMA is determined by the effective dimensionality of the convolutional code used for shaping. Forney points out that the effective dimensionality is longer than the code's constraint length, as evidenced by the fact that the trellis search will select different paths as the search window length is increased to many times the constraint length of the code [8].

CHAPTER 3

THE FRACTIONALLY SPACED VECTOR CONSTANT MODULUS ALGORITHM

3.1 The Constant Modulus Algorithm (CMA)

Consider the complex baseband model of a digital communication system shown in Figure 3.1. The transmitted data sequence $\{a_n\}$ consists of independent, identically distributed symbols chosen from a signal constellation with symmetries satisfying $E[a_n^2] = 0$. The transmitted sequence is filtered by a channel \mathbf{h} and passed through an equalizer \mathbf{c} . The equalizer output is $z = \mathbf{y} * \mathbf{c}$, where \mathbf{y} is a vector of samples of the channel output $y = \mathbf{a} * \mathbf{h}$.

Suppose the constant modulus algorithm (CMA) is used to adapt the equalizer [1], [2]. The cost function for CMA is

$$CF_{\text{CMA}} = E(|z_n|^p - R_p)^2 \quad (3.1)$$

where

$$R_p = \frac{E|a_n|^{2p}}{E|a_n|^p} \quad (3.2)$$

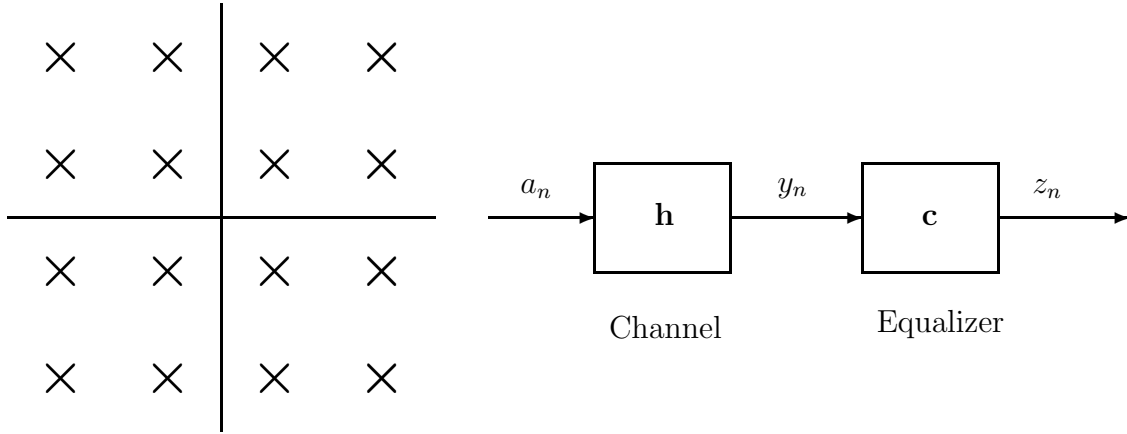


Figure 3.1 Communications system model. A sequence of symbols $\{a_n\}$ from a set in the complex plane (16-QAM is shown here) are convolved first with the channel impulse response \mathbf{h} and then with the equalizer \mathbf{c} at the receiver.

is a constant which depends on the signal constellation and acts as a scaling factor for the equalized data. As was mentioned previously, this cost function measures the deviation of the received signal points from a circle of constant modulus.

A stochastic gradient update [9] with step size μ is used to update the filter coefficients:

$$\mathbf{c}_{n+1} = \mathbf{c}_n - \mu \left[\frac{\partial C F_{\text{CMA}}}{\partial \mathbf{c}_n^*} \right] \quad (3.3)$$

where, by convention, the derivative is with respect to \mathbf{c}_n^* . Recalling that $z_n = \mathbf{y}_n^T \mathbf{c}_n$, this is calculated as

$$\begin{aligned} \left[\frac{\partial C F_{\text{CMA}}}{\partial \mathbf{c}_n^*} \right] &= \\ E \left[2(|z_n|^p - R_p) p |z_n|^{p-1} \frac{\partial}{\partial \mathbf{c}_n^*} ((\mathbf{y}_n^T \mathbf{c}_n)^* (\mathbf{y}_n^T \mathbf{c}_n))^{1/2} \right] &= \\ E \left[2(|z_n|^p - R_p) p |z_n|^{p-1} \frac{1}{2} |z_n|^{-1} \mathbf{y}_n^* z_n \right] & \quad (3.4) \end{aligned}$$

Dropping the expectation and simplifying, we obtain the desired tap update equation

$$\mathbf{c}_{n+1} = \mathbf{c}_n - \lambda \mathbf{y}_n^* z_n |z_n|^{p-2} (|z_n|^p - R_p) \quad (3.5)$$

where the step size λ is μ multiplied by the constant factors arising in the differentiation.

For computational convenience, p is normally set to 2.

3.2 Baud-Spaced VCMA

The vector constant modulus algorithm (VCMA) is an extension of CMA which operates on vectors of the received signal samples. For the reader's convenience, the framework and derivations for the baud-spaced vector constant modulus algorithm are reproduced here. See [5] and [10] for a longer discussion.

Let $\mathbf{a}_n = [a_n \ a_{n-1} \ \cdots \ a_{n-N+1}]^T$ be a vector of N transmitted complex signal points corresponding to a point in the $2N$ -dimensional uniform constellation (shell-mapping is in mind here). Let $\mathbf{z}_n = [z_n \ z_{n-1} \ \cdots \ z_{n-N+1}]^T$ be a vector of the last N output samples from the equalizer; it is calculated by $\mathbf{z}_n = \mathbf{Y}_n^T \mathbf{c}_n$, where

$$\mathbf{Y}_n = \begin{bmatrix} y_n & y_{n-1} & \cdots & y_{n-N+1} \\ y_{n-1} & y_{n-2} & \cdots & y_{n-N} \\ \vdots & \vdots & & \vdots \\ y_{n-L_c+1} & y_{n-L_c} & \cdots & y_{n-L_c-N+2} \end{bmatrix} \quad (3.6)$$

and L_c is the number of equalizer taps. Note that here we have implicitly assumed a baud-spaced equalizer; a fractionally-spaced example is given in the following section.

VCMA uses a cost function identical to that of CMA except that a vector modulus is used:

$$CF_{\text{VCMA}} = E(|\mathbf{z}_n|^p - R_p)^2 \quad (3.7)$$

where

$$R_p = \frac{E|\mathbf{a}_n|^{2p}}{E|\mathbf{a}_n|^p} \quad (3.8)$$

We again use a stochastic gradient update with step size μ to update the filter coefficients:

$$\mathbf{c}_{n+1} = \mathbf{c}_n - \mu \left[\frac{\partial CF_{\text{VCMA}}}{\partial \mathbf{c}_n^*} \right] \quad (3.9)$$

Recalling that $\mathbf{z}_n = \mathbf{Y}_n^T \mathbf{c}_n$, the derivative is calculated as

$$\begin{aligned} \left[\frac{\partial CF_{\text{VCMA}}}{\partial \mathbf{c}_n^*} \right] &= \\ E \left[2(|\mathbf{z}_n|^p - R_p)p|\mathbf{z}_n|^{p-1} \frac{\partial}{\partial \mathbf{c}_n^*} ((\mathbf{Y}_n^T \mathbf{c}_n)^H (\mathbf{Y}_n^T \mathbf{c}_n))^{1/2} \right] &= \\ E \left[2(|\mathbf{z}_n|^p - R_p)p|\mathbf{z}_n|^{p-1} \frac{1}{2} |\mathbf{z}_n|^{-1} \mathbf{Y}_n^* \mathbf{z}_n \right] & \quad (3.10) \end{aligned}$$

Dropping the expectation and simplifying, we obtain the desired tap update equation

$$\mathbf{c}_{n+1} = \mathbf{c}_n - \lambda \mathbf{Y}_n^* \mathbf{z}_n |\mathbf{z}_n|^{p-2} (|\mathbf{z}_n|^p - R_p) \quad (3.11)$$

Table 3.1 Computational steps and their associated costs for one iteration of VCMA.

Operation	Real multiplications	Real additions
1. $\mathbf{u} = \mathbf{y}_N$		
2. ⟨Update \mathbf{Y} — see text⟩		
3. $v = \mathbf{y}_1^T \mathbf{c}$	$4L_c$	$4L_c - 2$
4. $b = b + v ^2 - z_N ^2$	2	3
5. $e = b - R_2$		1
6. $\mathbf{x} = \mathbf{x} + \mathbf{y}_1^* v - \mathbf{u}^* z_N$	$8L_c$	$8L_c$
7. $\mathbf{z} = [v \ z_1 \ z_2 \ \cdots \ z_{N-1}]^T$		
8. $\mathbf{c} = \mathbf{c} - \lambda \mathbf{x} e$	$2L_c + 1$	$2L_c$
Total:	$14L_c + 3$	$14L_c + 2$

$$\left(\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_N \end{bmatrix} \quad ; \quad \mathbf{z} = [z_1 \ z_2 \ \cdots \ z_N]^T \right)$$

where the step size λ is μ multiplied by the constant factors arising in the differentiation.

As with CMA, p is normally set to 2 for computational convenience. The equalizer taps are updated every symbol period. Under these conditions, the computational cost of VCMA is surprisingly low. If all data values are kept in memory between tap updates, several computational short-cuts are possible. For many of these simplifications it is necessary to assume that the tap weights \mathbf{c}_n change slowly over time so that \mathbf{c}_n is an acceptable substitute for \mathbf{c}_{n-m} , $0 < m < N$.

The suggested steps of computation and their cost in real multiplications and additions are shown in Table 3.1. It will be necessary to run the recursive update steps through at least N iterations to properly initialize the matrices and vectors before starting filter adaptation. Following are some notes on the calculations:

1. The last column of \mathbf{Y} is saved for later use.

2. The matrix \mathbf{Y} is updated with new data by shifting each row to the right and discarding the last column. The first column is updated by shifting it down m elements for a T/m -spaced equalizer (see the next section), then filling in the top m positions with new data.
3. The new first element of \mathbf{z} is calculated, requiring L_c complex multiplications and $L_c - 1$ complex additions.
4. The vector norm of \mathbf{z} is denoted b , and is recursively updated here by adding the squared magnitude of the new first element and subtracting the squared magnitude of the old last element. The latter quantity was computed N iterations earlier and can be stored in memory so that only one magnitude need be calculated. The cost is two real multiplications and one real addition for the magnitude calculation, plus two real additions to find the new b .
5. The error e is real and requires just one real addition to calculate.
6. The vector \mathbf{x} represents the recursively updated product $\mathbf{Y}^*\mathbf{z}$. Two column/scalar multiplies are used—the first and last columns of \mathbf{Y} with the new first element and the old last element of \mathbf{z} , respectively, requiring $2L_c$ complex multiplications. Then, a column addition and subtraction are performed using $2L_c$ complex additions.
7. In preparation for the next iteration, \mathbf{z} is updated by shifting the elements down, discarding the last one, and placing the previously computed v in the first position.

8. Both λ and e are real, so the filter update equation now requires only one real multiplication to find λe , $2L_c$ real multiplications to multiply the complex column vector \mathbf{x} by this a real number, and L_c complex additions to update the filter taps **c**.

The end result is a cost of $14L_c + 3$ real multiplications and $14L_c + 2$ real additions per symbol period, which is only about 75% more expensive than conventional CMA. Considering the transformation of the update equation from a scalar/vector form (CMA) to a vector/matrix form (VCMA), this is a pleasant surprise!

“Decimating the update” by updating the taps less frequently than every symbol period does not deliver the computational savings that might be expected, since some of the savings offered by the recursive updates are forfeited. For example, if the update is computed only every other symbol period, there is an inherent halving of the required computations (or so it would seem at first glance), but this is almost entirely offset because steps 3, 4, and 6 (above) now require twice as many computations.

It is also worth noting that the computational cost of VCMA is virtually independent of the block size N when implemented with the speed-ups discussed here. In Chapter 6 it will be shown that N may often be set lower than the theoretical requirement while preserving good performance. This does not, however, lead to a significant computational gain.

3.3 Fractionally Spaced VCMA

It is often desirable to use an equalizer with taps spaced at some fraction of the data symbol period T , most commonly $T/2$. A fractionally spaced equalizer (FSE), as this configuration is termed, has the extra degrees of freedom necessary to perform additional receiver operations such as matched filtering and adjustment of sampling phase. This makes for simpler, better receivers. More recently it was shown that while an ordinary baud-spaced equalizer of arbitrary length cannot perfectly equalize a general FIR channel, a CMA FSE of length equal to or greater than the channel delay spread can achieve global convergence given some simple channel restrictions [11]. A “good” FSE is therefore often shorter than a “good” baud-spaced equalizer (BSE) for a given channel. For these reasons, fractionally-spaced equalizers are now the norm in most applications.

In a FSE, the channel is sampled at the desired multiple of the symbol rate and the equalizer output is calculated only at T -spaced intervals to obtain the equalized data. Clearly, when an adaptive algorithm like CMA is used with a FSE, the tap update operation should be performed not more frequently than at time intervals of T to avoid over-constraining the filter output, since only the decimated rate $1/T$ equalizer output is of interest. When applied to VCMA, this requires some simple changes in the structure of the computations. Recall that the equalizer output vector is $\mathbf{z}_n = \mathbf{Y}_n^T \mathbf{c}_n$, with the rectangular matrix \mathbf{Y}_n given by (3.6) for the baud-spaced case. For a $T/2$ -spaced FSE, the samples entering the equalizer need to be $T/2$ -spaced, so each column of \mathbf{Y}_n should be $T/2$ -spaced. The equalizer output vector \mathbf{z}_n , however, needs to be T -spaced, so each

row of \mathbf{Y}_n should be T -spaced. Thus, for a $T/2$ -spaced FSE, we have

$$\mathbf{Y}_n = \begin{bmatrix} y_n & y_{n-2} & \cdots & y_{n-2N+2} \\ y_{n-1} & y_{n-3} & \cdots & y_{n-2N+1} \\ \vdots & \vdots & & \vdots \\ y_{n-L_c+1} & y_{n-L_c-1} & \cdots & y_{n-L_c-2N+3} \end{bmatrix} \quad (3.12)$$

Simulations performed with FS-VCMA and presented in Chapter 6 have verified its ability to equalize several short FIR channels with a required equalizer length comparable to the channel delay spread.

Analyzing the computational cost of FS-VCMA in like manner to that of the baud-spaced case, one finds that the required steps in calculating the update are the same. The only change in the structure of the calculations is the added space between columns of \mathbf{Y}_n , and this does not affect any of the shortcuts. Therefore, FS-VCMA still requires $14L_c + 3$ real multiplications and $14L_c + 2$ real additions *per symbol period* T . A real-life analysis would, of course, have to account for the change in required L_c when moving from a BSE to a FSE.

3.4 Normalized Fractionally Spaced VCMA

In applications where convergence speed is a limiting factor in performance, such as when the symbol rate is low, it may be desirable to use a version of VCMA with a

normalized step size λ . This was first derived for baud-spaced VCMA by Yang in [10], and is reproduced here for the $T/2$ fractionally-spaced case.

The *a posteriori* error

$$\varepsilon_n = |\mathbf{Y}_n^T \mathbf{c}_{n+1}|^2 - R_2 \quad (3.13)$$

is the error that would be obtained using the following iteration's equalizer tap vector on the current iteration's data. By using the known tap-update equation to substitute for \mathbf{c}_{n+1} , this instantaneous *a posteriori* error can be set to zero to solve for a step size λ that will then be applied in the tap update calculation:

$$\varepsilon_n = |\mathbf{Y}_n^T (\mathbf{c}_n - \lambda \mathbf{Y}_n^* \mathbf{z}_n e_n)|^2 - R_2 \quad (3.14)$$

$$= |\mathbf{z}_n - \lambda \mathbf{Y}_n^T \mathbf{Y}_n^* \mathbf{z}_n e_n|^2 - R_2 \quad (3.15)$$

$$= e_n - 2 \mathbf{z}_n^H \mathbf{Y}_n^T \mathbf{Y}_n^* \mathbf{z}_n e_n \lambda + \mathbf{z}_n^H \mathbf{Y}_n^T \mathbf{Y}_n^* \mathbf{Y}_n^T \mathbf{Y}_n^* \mathbf{z}_n e_n^2 \lambda^2 \quad (3.16)$$

$$= e_n - 2 \mathbf{z}_n^H \mathbf{Y}_n^T \mathbf{Y}_n^* \mathbf{z}_n e_n \lambda + |\mathbf{Y}_n^T \mathbf{Y}_n^* \mathbf{z}_n|^2 e_n^2 \lambda^2 \quad (3.17)$$

where $e_n = |\mathbf{Y}_n^T \mathbf{c}_n|^2 - R_2$. The coefficients of this quadratic equation are real and may be computed as follows.

The factor $\mathbf{Y}_n^T \mathbf{Y}_n^*$ can be written as

$$\mathbf{Y}_n^T \mathbf{Y}_n^* = \begin{bmatrix} r_n(0) & r_n(2) & \cdots & r_n(2N-2) \\ r_n^*(2) & r_{n-2}(0) & \cdots & r_{n-2}(2N) \\ \vdots & \vdots & & \vdots \\ r_n^*(2N-2) & r_{n-2}^*(2N) & \cdots & r_{n-2N+2}(0) \end{bmatrix} \quad (3.18)$$

where

$$r_l(j) = \sum_{k=0}^{L_c-1} y_{l-k} y_{l-k-j}^* \quad (3.19)$$

Note the relation

$$\begin{aligned} r_{l+2}(j) &= \sum_{k=0}^{L_c-1} y_{l-(k-2)} y_{l-(k-2)-j}^* \\ &= r_l(j) + y_{l+1} y_{l+1-j}^* + y_{l+2} y_{l+2-j}^* - y_{l-L_c+1} y_{l-L_c+1-j}^* - y_{l-L_c+2} y_{l-L_c+2-j}^* \end{aligned} \quad (3.20)$$

which allows the matrix to be updated recursively. All elements except the top row and left column are obtained by shifting down and right. Taking advantage of the hermitian symmetry, the remaining elements can be computed with $4N$ complex multiplications and $4N$ complex additions.

To obtain the coefficient of the λ^2 term, we multiply the foregoing matrix product by \mathbf{z}_n , take the magnitude squared, and multiply by e_n^2 . To multiply by \mathbf{z}_n , the first element is computed from scratch with N complex multiplications and $N - 1$ complex additions. The remaining $N - 1$ elements are recursively calculated using $2(N - 1)$ complex multiplications and $2(N - 1)$ complex additions. The magnitude squared operation requires $2N$ real multiplications and $2(N - 1)$ real additions. The error e_n was already calculated, so two real multiplications are needed to square it and multiply with the other result.

To obtain the coefficient of the λ term, we multiply the already-computed quantity $\mathbf{Y}_n^T \mathbf{Y}_n^* \mathbf{z}_n$ by \mathbf{z}_n^H and, since the result is real, this requires $2N$ real multiplications and $2N - 1$ real additions. Multiplying by $2e_n$ contributes another two real multiplications.

The total computational cost for one iteration is $32N - 4$ real multiplications and $32N - 13$ real additions, plus the extra computations needed to solve the quadratic equation. After obtaining the two possible values for λ , the smallest positive root should be chosen.

Since VCMA is typically used in conjunction with large, dense, nonconstant-modulus signal constellations, λ will remain very noisy after convergence and this excess noise may prevent a switch to decision-directed equalization. To overcome this problem, it is recommended that the “optimal” λ be scaled by some empirically-determined factor in the tap update equation. This allows a compromise between slower convergence and equalizer noise.

CHAPTER 4

CONVERGENCE PROOFS BASED ON KURTOSIS CRITERIA

Much of the recent work on blind equalization algorithms has focused on these algorithms' convergence properties or the lack thereof. The global convergence of baud-spaced CMA to a perfectly equalizing filter was shown by Foschini [12] for the impractical case of a doubly-infinite equalizer in the absence of noise. Foschini's proof was enabled by working in the combined channel-equalizer parameter space (described in more detail below), thus necessitating an infinitely long equalizer to allow reaching every point in this space. Other results have documented the misconvergence of CMA when the equalizer is not infinitely long [13]. More recently, Li and Ding [11] were able to prove the global convergence of fractionally-spaced CMA for finite-length equalizers, provided the equalizer has a length greater than the channel delay spread and the subchannels (polyphase decomposition of the channel) do not have common zeros.

One would hope that convergence properties similar to these would hold for baud- and fractionally spaced VCMA, but the straightforward application of the same proof techniques to VCMA has been only partially successful. Using a proof similar to Foschini's, Yang [10] was able to show convergence of VCMA to a setting with N contiguous nonzero

filter taps in the combined channel-equalizer parameter space (perfect equalization would imply only one nonzero filter tap).

The vector nature of the algorithm introduces many additional cross terms that complicate the mathematics enormously. Because of this, it was decided to try a fresh approach to the problem using a kurtosis-based viewpoint of CMA which might be more easily extended to VCMA than the earlier methods. Here the kurtosis of a random variable X is defined as

$$\kappa_X = \frac{E[|X|^4]}{(E[|X|^2])^2} \tag{4.1}$$

A kurtosis-based analysis of the blind deconvolution problem has also been performed by Shalvi and Weinstein in [14]. In their work, however, a different definition is used for the kurtosis, and the focus is on the development of a new algorithm which can equalize any non-Gaussian (sub-Gaussian or super-Gaussian) source distribution.

As shown in the next section, one can think of the CMA cost function as a kurtosis-minimizing criterion. Basic results about the kurtosis of sums of random variables may then be used to show that CMA attains perfect equalization (note that an infinite-length equalizer is still assumed here). This has the added benefit of showing simply and intuitively how the constant modulus criterion reduces intersymbol interference and equalizes the channel. Unfortunately, it turns out that the kurtosis of a sum of random vectors does not share the same useful properties for our purposes as in the scalar case; therefore, this proof has not been extended to VCMA.

4.1 Kurtosis in the CMA Cost Function

Consider the equalizer model shown in Figure 4.1. The filter \mathbf{c} represents the channel-equalizer cascade and is normalized to unit energy by the constraint

$$\sum_n |c_n|^2 = 1 \quad (4.2)$$

This filter is followed by a real, scalar gain constant α . Letting $R_2 = E|z|^4/E|z|^2$, the CMA cost function becomes

$$CF_{\text{CMA}} = E \left[(|z|^2 - R_2)^2 \right] = E \left[(\alpha^2 |v|^2 - R_2)^2 \right] \quad (4.3)$$

Now solve for the optimal α , denoted by α_* :

$$\min_{\alpha} \left(\alpha^4 E|v|^4 - 2\alpha^2 R_2 E|v|^2 + R_2^2 \right) \quad (4.4)$$

$$\frac{\partial CF_{\text{CMA}}}{\partial \alpha} = 4\alpha^3 E|v|^4 - 4\alpha R_2 E|v|^2 = 0 \quad (4.5)$$

$$\alpha_* = \left(\frac{R_2 E|v|^2}{E|v|^4} \right)^{1/2} \quad (4.6)$$

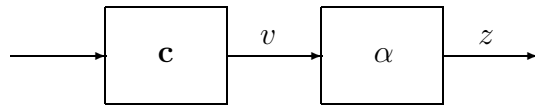


Figure 4.1 Equalizer model showing the partition into a unit-energy vector \mathbf{c} and a scalar gain factor α .

Note that although $\alpha_* = 0$ is a solution, it is not optimal, but represents the trivial solution of an equalizer with all taps equal to zero. Substitute the optimal gain back into the cost function

$$\min_{\mathbf{c}} E \left[\left(\frac{R_2 |v|^2 E |v|^2}{E |v|^4} - R_2 \right)^2 \right] \quad (4.7)$$

$$\min_{\mathbf{c}} E \left[\left(\frac{|v|^2 E |v|^2}{E |v|^4} - 1 \right)^2 \right] \quad (4.8)$$

$$\min_{\mathbf{c}} \left(\frac{(E |v|^2)^2 E |v|^4}{(E |v|^4)^2} - \frac{2E |v|^2 E |v|^2}{E |v|^4} + 1 \right) \quad (4.9)$$

$$\min_{\mathbf{c}} \left(1 - \frac{(E |v|^2)^2}{E |v|^4} \right) \quad (4.10)$$

$$\min_{\mathbf{c}} \left(1 - \frac{1}{\kappa_v} \right) \quad (4.11)$$

where κ_v is the kurtosis of the data at the output of the normalized filter and is bounded below by one (corresponding to constant data). Complex Gaussian data would have a kurtosis of two by our definition. Thus, the central limit theorem sets an upper bound of two for κ_v if the transmitted data has a sub-Gaussian distribution (this will be required in the subsequent derivations).

It is easy to verify that the kurtosis is independent of the gain α and thus $\kappa_v = \kappa_z$, the kurtosis of the equalizer output data. This shows that minimizing the standard CMA cost function is equivalent to simultaneously minimizing the kurtosis of the equalized data and optimizing the overall gain α . This also helps explain why CMA begins to fail as the source data undergoes progressively more shaping at the transmitter, with

marginal distributions becoming closer and closer to Gaussian—the kurtosis starts to become “irreducible.”

4.2 Kurtosis of a Linear Combination of Random Variables

Now consider the kurtosis of a linear sum of random variables with complex-valued coefficients. Specifically, start with two independent, complex random variables X_1 and X_2 , not necessarily chosen from the same distribution, and their weighted sum

$$S = b_1X_1 + b_2X_2 \quad (4.12)$$

where b_1 and b_2 are complex in general. The kurtosis of this linear combination is

$$\kappa_s = \frac{E|b_1X_1 + b_2X_2|^4}{(E|b_1X_1 + b_2X_2|^2)^2} = \frac{A_1|b_1|^4 + b_1^2b_2^{*2}C_1C_2^* + 4|b_1|^2|b_2|^2B_1B_2 + b_1^{*2}b_2^2C_1^*C_2 + A_2|b_2|^4}{|b_1|^4B_1^2 + 2|b_1|^2|b_2|^2B_1B_2 + |b_2|^4B_2^2} \quad (4.13)$$

where the constants

$$\begin{aligned} A_1 &= E|X_1|^4 & A_2 &= E|X_2|^4 \\ B_1 &= E|X_1|^2 & B_2 &= E|X_2|^2 \\ C_1 &= E[X_1^2] & C_2 &= E[X_2^2] \end{aligned} \quad (4.14)$$

have been introduced for more compact notation. To find the extrema of κ_s we take the partial derivative with respect to b_1 or b_2 . It will be seen that given any constant b_2 (respectively b_1), the kurtosis is minimized by setting b_1 (respectively b_2) equal to zero. Therefore, for any $b_1 \neq 0$, $b_2 \neq 0$, a lower kurtosis κ_s could be obtained by setting either

$b_1 = 0$ or $b_2 = 0$. The single partial derivatives then suffice to jointly minimize κ_S with respect to b_1 and b_2 .

Taking the derivative with respect to b_1 , setting the numerator equal to zero, and simplifying, we obtain

$$2B_1B_2b_1^*|b_2|^2 \left[(A_1 - 2B_1^2)|b_1|^4 - (A_2 - 2B_2^2)|b_2|^4 \right] + 2|b_1|^2b_1^*|b_2|^4(A_1B_2^2 - A_2B_1^2) + 2 \left(B_1|b_1|^2 + B_2|b_2|^2 \right) \left(B_2C_1C_2^*b_1|b_2|^2b_2^{*2} - B_1C_1^*C_2b_1^{*3}b_2^2 \right) = 0 \quad (4.15)$$

(Taking the derivative with respect to b_2 leads to the same equation with 1's and 2's switched due to the symmetry in κ_S .)

Extensive use has been made here of the complex differentiation identities [9], [15]

$$\frac{\partial}{\partial b} = \frac{1}{2} \left(\frac{\partial}{\partial \mathcal{R}e\{b\}} - j \frac{\partial}{\partial \mathcal{I}m\{b\}} \right) \quad (4.16)$$

$$\frac{\partial |b|^2}{\partial b} = b^* \quad (4.17)$$

$$\frac{\partial b^*}{\partial b} = 0 \quad (4.18)$$

where the last two identities follow from the first, and $\mathcal{R}e\{\cdot\}$ and $\mathcal{I}m\{\cdot\}$ signify the real and imaginary parts, respectively.

Assume that X_1 and X_2 are sub-Gaussian, so that $\kappa_{X_1}, \kappa_{X_2} < 2$. Also require $C_1 = C_2 = 0$, which will be true for most complex signal constellations such as quadrature amplitude modulation (QAM). Then there are only two possible conditions for extrema

of κ_S ; either

$$1) \quad b_1 = 0, \text{ or } b_2 = 0$$

or

$$2) \quad B_1 B_2 \left[(A_1 - 2B_1^2)|b_1|^4 - (A_2 - 2B_2^2)|b_2|^4 \right] + |b_1|^2 |b_2|^2 (A_1 B_2^2 - A_2 B_1^2) = 0$$

Given the first condition, the Hessian is

$$\begin{bmatrix} \frac{\partial^2 \kappa_S}{\partial b_1^* \partial b_1} & \frac{\partial^2 \kappa_S}{\partial b_1^* \partial b_2} \\ \frac{\partial^2 \kappa_S}{\partial b_2^* \partial b_1} & \frac{\partial^2 \kappa_S}{\partial b_2^* \partial b_2} \end{bmatrix} = \begin{bmatrix} 2B_1 B_2 (A_2 - 2B_2^2) |b_2|^6 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 2B_1 B_2^3 (2 - \kappa_{X_2}) |b_2|^6 & 0 \\ 0 & 0 \end{bmatrix} \quad (4.19)$$

if $b_1 = 0$ or

$$\begin{bmatrix} 0 & 0 \\ 0 & 2B_1^3 B_2 (2 - \kappa_{X_1}) |b_1|^6 \end{bmatrix} \quad (4.20)$$

if $b_2 = 0$, where we have made use of the fact that kurtosis $\kappa = A/B^2$ given the previous definitions. In either case the eigenvalues are always greater than or equal to zero, for κ_{X_1} and $\kappa_{X_2} < 0$. Thus, the first condition represents a local minimum or saddle point for sub-Gaussian processes.

If we assume condition (2), the Hessian becomes

$$2B_1 B_2 \left[(A_1 - 2B_1^2)|b_1|^4 + (A_2 - 2B_2^2)|b_2|^4 \right] \begin{bmatrix} -|b_2|^2 & b_1 b_2^* \\ b_1^* b_2 & -|b_1|^2 \end{bmatrix} \quad (4.21)$$

$$= 2B_1B_2 \left[(2 - \kappa_{x_1})B_1^2|b_1|^4 + (2 - \kappa_{x_2})B_2^2|b_2|^4 \right] \begin{bmatrix} -|b_2|^2 & b_1b_2^* \\ b_1^*b_2 & -|b_1|^2 \end{bmatrix} \quad (4.22)$$

where each factor of the form $A-2B^2$ has again been rewritten in terms of a kurtosis. This matrix has eigenvalues 0 and $-2B_1B_2(|b_1|^2 + |b_2|^2)[(2 - \kappa_{x_1})B_1^2|b_1|^4 + (2 - \kappa_{x_2})B_2^2|b_2|^4]$, both of which are less than or equal to zero under the stated assumptions. Therefore, the second condition represents a local maximum or saddle point.

The kurtosis κ_S is defined on an unbounded domain. Its value is bounded below by one. Therefore, the minimum noted above must be the global minimum for κ_S ; in particular, κ_S can be minimized only by setting either b_1 or b_2 to zero.

Because there was no requirement for X_1 and X_2 to be identically distributed, this proof can be extended to any number of variables by induction. Let P_N be the following proposition:

The kurtosis of a linear sum, with complex coefficients, of $N + 1$ complex, independent, but not necessarily identically distributed random variables X_n is globally minimized by a set of coefficients having the form $[0 \cdots 0 \ b \ 0 \cdots 0]$, given that each random variable has a complex kurtosis less than two and symmetries such that $E[X_n^2] = 0$.

P_1 was proved in the preceding derivation. If we assume that P_N holds, we see that $P_N \Rightarrow P_{N+1}$ for all N since a “new” random variable can be added to any one of the other random variables without violating the conditions of P_N . Thus, by induction, the proposition holds for all N .

4.3 A Proof of Convergence for CMA

These derivations can be used to construct a kurtosis-based proof of global convergence for CMA in a noise-free setting with an infinite-length equalizer as follows. For this case it is convenient to work in the combined channel-equalizer space where, following Foschini [12], we consider one set of filter taps representing the cascade of the channel and the equalizer. For any given channel, every point in the combined channel-equalizer space, including the perfect equalization case, may then be reached with some setting of the equalizer taps. This is only possible because the equalizer is assumed to be infinitely long; if it is not, then the mapping between equalizer space and combined channel-equalizer space is not bijective and there is no easy guarantee that a perfectly-equalizing setting of the equalizer taps exists.

Part one of the derivation (Eqs. (4.1) through (4.9)) shows that if the CMA cost function is minimized, the output kurtosis must also be minimized. Part two is the previous section, which shows that the output kurtosis can only be minimized by attaining perfect equalization, which is defined here as a combined channel-equalizer unit pulse response having a single nonzero element. Combining these results, it can be stated that if perfect equalization has not been attained, then the CMA cost function has not been minimized.

Now consider the possibility of local minima in the CMA cost function. If the equalizer gain α is not optimal, i.e., $\alpha \neq \alpha_*$, then that point in the tap space cannot be a local minimum since a better α would reduce the cost. It is also clear from the previous

section that the unconstrained minimization of the output kurtosis over the combined channel-equalizer space has no local minima. Thus, neither does the CMA cost function have any local minima, because the output kurtosis does not depend on the gain α .

Originally it was hoped that this approach to a CMA convergence proof could be used to obtain a proof of convergence for VCMA, but we have been unable to do so. If length- N random vectors are substituted for the random variables in Equation (4.12), the result is a larger total number of noncombinable terms in the expression for κ_s , Equation (4.13). Due to these extra terms, it is not possible to identify the individual kurtosis factors in the resulting expression and the desired result remains unproven.

4.4 Constellations for which $E[X^2] \neq 0$

For some signal constellations, most notably pulse amplitude modulation (PAM), $C = E[X^2] \neq 0$ and there may be other channel-equalizer pulse responses which minimize kurtosis but do not correspond to perfect equalization. To gain some insight, consider the simple two-tap case (b_1 and b_2) with identical distributions for X_1 and X_2 . Simplifying the derivative of κ_s from Equation (4.15) yields the condition

$$2b_1^*|b_2|^2 B^2(A - 2B^2) \left[|b_1|^4 - |b_2|^4 \right] + 2B^2|C|^2 \left(|b_1|^2 + |b_2|^2 \right) \left[b_1 b_2^{*2} |b_2|^2 - b_1^{*3} b_2^2 \right] = 0 \quad (4.23)$$

Let $b_1 = r_1 e^{j\theta_1}$ and $b_2 = r_2 e^{j\theta_2}$. The rightmost bracketed factor becomes

$$r_1 e^{j\theta_1} r_2^4 e^{-j2\theta_2} = r_1^3 e^{-j3\theta_1} r_2^2 e^{j2\theta_2} \quad (4.24)$$

Notice that the terms in brackets can both equal zero only when b_1 and b_2 have a common magnitude $r_1 = r_2 = r$ with the angles related by

$$\theta_1 - 2\theta_2 = -3\theta_1 + 2\theta_2 + 2\pi n \quad (4.25)$$

$$\theta_1 = \theta_2 + \frac{\pi n}{2} \quad (4.26)$$

where n is an integer. Thus, even in this simple case there are many extrema of κ_S which do not correspond to perfect equalization. Some or all of these points in the channel-equalizer space probably correspond to undesirable minima of the kurtosis. In fact, this has been shown to be the case by Papadias [16], who considered the case of binary PAM signals. The results in [16] agree very well with this simple example.

CHAPTER 5

GENERALIZING THE VECTOR MODULUS CONCEPT TO OTHER ALGORITHMS

The constant modulus algorithm can be viewed as just one member of a larger family of blind equalization algorithms having similar cost functions and typically implemented with a stochastic gradient algorithm. For example, the reduced constellation algorithm (RCA) proposed in [17] uses the cost function

$$CF_{\text{RCA}} = E|z_n - Rcsgn(z_n)|^2 \quad (5.1)$$

where $csgn$ is the complex sign function which returns one of the values in $\{1 + j, 1 - j, -1 + j, -1 - j\}$ according to the quadrant occupied by z_n . This cost function attempts to fit the received constellation points to the corners of a square. Although reportedly more prone to misconvergence than other algorithms in the same family, RCA is simple to implement and has the advantage of correcting for carrier phase, since unlike CMA the cost function is sensitive to constellation rotation [18]. In this chapter, some preliminary experiments with vector versions of this and other algorithms are considered.

5.1 Vector Reduced Constellation Algorithm

A vector RCA (VRCA) is easily derived by using vector quantities in place of the scalar quantities in the cost function:

$$CF_{\text{VRCA}} = E|\mathbf{z}_n - R\text{csgn}(\mathbf{z}_n)|^2 \quad (5.2)$$

The complex sign function csgn returns a vector the same length as \mathbf{z} , working on an element-by-element basis. Whereas we can think of VCMA attempting to fit vectors of received symbols to a hypersphere in $2N$ dimensions, VRCA attempts to fit these vectors to the corners of a $2N$ -dimensional hypercube. The update equation for VRCA (after dropping the expectation) is

$$\begin{aligned} \mathbf{c}_{n+1} &= \mathbf{c}_n - \lambda \left[\frac{\partial CF_{\text{VRCA}}}{\partial \mathbf{c}_n^*} \right] \\ &= \mathbf{c}_n - \lambda \frac{\partial}{\partial \mathbf{c}_n^*} [(\mathbf{c}_n^H \mathbf{Y}_n^* - R\text{csgn}(\mathbf{c}_n^H \mathbf{Y}_n^*))(\mathbf{Y}_n^T \mathbf{c}_n - R\text{csgn}(\mathbf{Y}_n^T \mathbf{c}_n))] \\ &= \mathbf{c}_n - \lambda \mathbf{Y}_n^* (\mathbf{z}_n - R\text{csgn}(\mathbf{z}_n)) \end{aligned} \quad (5.3)$$

Computationally, VRCA is very similar to VCMA. If all data values are kept in memory between tap updates, and given the same assumption of a slowly-varying \mathbf{c}_n used for the VCMA simplifications (see Chapter 3), several computational short-cuts are possible. For an equalizer of length L_c , the suggested sequence of operations and the corresponding costs are illustrated in Table 5.1. The following comments refer to the numbered steps in the figure:

Table 5.1 Computational steps and their associated costs for one iteration of VRCA.

Operation	Real multiplications	Real additions
1. $\mathbf{u} = \mathbf{y}_N$		
2. $\langle \text{Update } \mathbf{Y} \rangle$		
3. $v = \mathbf{y}_1^T \mathbf{c}$	$4L_c$	$4L_c - 2$
4. $t = v - R\text{csgn}(v)$		2
5. $\mathbf{x} = \mathbf{x} + \mathbf{y}_1^* t - \mathbf{u}^* e_N$	$8L_c$	$8L_c$
7. $\mathbf{e} = [t \ e_1 \ e_2 \ \cdots \ e_{N-1}]^T$		
8. $\mathbf{c} = \mathbf{c} - \lambda \mathbf{x}$	$2L_c$	$2L_c$
Total:	$14L_c$	$14L_c$

$$\left(\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_N \end{bmatrix} \ ; \ \mathbf{e} = [e_1 \ e_2 \ \cdots \ e_N]^T \right)$$

1. The last column of \mathbf{Y} is saved for later use.
2. The matrix \mathbf{Y} is updated with new data by shifting each row to the right and discarding the last column. The first column is updated by shifting it down m elements for a T/m -spaced equalizer, then filling in the top m positions with new data.
3. The new first element of \mathbf{z} is denoted v and is computed with L_c complex multiplications and $L_c - 1$ complex additions.
4. The new first element of the error vector $\mathbf{e} = \mathbf{z} - R\text{csgn}(\mathbf{z})$ is t . Since there are only four possible values of $R\text{csgn}(v)$, it is assumed here that the only calculation required to find t is one complex addition.
5. The vector x represents the recursively updated product $\mathbf{Y}^*(\mathbf{z} - R\text{csgn}(\mathbf{z}))$. Two column/scalar multiplies are used—the first and last columns of \mathbf{Y} with the new first element and the old last element of the error vector $\mathbf{e} = \mathbf{z} - R\text{csgn}(\mathbf{z})$, respectively,

requiring $2L_c$ complex multiplications. Then, a column addition and subtraction are performed using $2L_c$ complex additions.

6. The filter update equation requires $2L_c$ real multiplications to multiply the complex column vector \mathbf{x} by the real scalar λ , then L_c complex additions to update the filter taps \mathbf{c} .

As with VCMA, the rate of computation required for fractionally-spaced VRCA is the same per symbol period as baud-spaced VRCA, and performing the tap update less frequently does not result in significant savings, due to the loss of computational speedups.

Note that there is no particular computational advantage to VRCA over VCMA. A principal advantage of RCA over CMA, its ability to correct a carrier phase offset, may or may not be a factor with VRCA depending on the type and degree of shaping used for the signal constellation. Shaped constellations are often sufficiently circular that VRCA will not be sensitive to constellation rotation. On the other hand, in trials with lightly trellis-shaped data, enough “squareness” remained for VRCA to correct for carrier phase offsets (see Chapter 6).

5.2 Vector Multimodulus Algorithm

Similar derivations and experiments were carried out with a vector version of the multimodulus algorithm (MMA) described in [19]. The simplest form of MMA has the

cost function

$$CF_{\text{MMA}} = E \left[(\mathcal{R}e^2\{z_n\} - R)^2 + (\mathcal{I}m^2\{z_n\} - R)^2 \right] \quad (5.4)$$

The update equation for MMA, found by taking the derivative with respect to \mathbf{c}_n^* and dropping the expectation, is

$$\begin{aligned} \mathbf{c}_{n+1} &= \mathbf{c}_n - \mu \left[\frac{\partial CF_{\text{MMA}}}{\partial \mathbf{c}_n^*} \right] \\ &= \mathbf{c}_n - \mu \frac{\partial}{\partial \mathbf{c}_n^*} \left[\left(\frac{1}{2} (\mathbf{y}_n^T \mathbf{c}_n + \mathbf{y}_n^H \mathbf{c}_n^*) \right)^2 - R \right]^2 + \left[\left(\frac{1}{2j} (\mathbf{y}_n^T \mathbf{c}_n - \mathbf{y}_n^H \mathbf{c}_n^*) \right)^2 - R \right]^2 \\ &= \mathbf{c}_n - \lambda \left[\mathcal{R}e\{z_n\} \mathbf{y}_n^* (\mathcal{R}e^2\{z_n\} - R) + j \mathcal{I}m\{z_n\} \mathbf{y}_n^* (\mathcal{I}m^2\{z_n\} - R) \right] \end{aligned} \quad (5.5)$$

where $\lambda = 2\mu$.

This is easily generalized to a vector MMA (VMMA) by making the z_n 's vector quantities. The resulting update equation is

$$\mathbf{c}_{n+1} = \mathbf{c}_n - \lambda \mathbf{Y}_n^* \left[\mathcal{R}e\{\mathbf{z}_n\} \left(|\mathcal{R}e\{\mathbf{z}_n\}|^2 - R \right) + j \mathcal{I}m\{\mathbf{z}_n\} \left(|\mathcal{I}m\{\mathbf{z}_n\}|^2 - R \right) \right] \quad (5.6)$$

Simulations yielded results much like those for VRCA, and similar comments apply.

As is apparent from these examples, the vector modulus concept from VCMA has general applicability and can be used to obtain other blind equalization algorithms by suitable modifications of the original cost function.

CHAPTER 6

SIMULATIONS

An extensive set of simulations was performed using fractionally-spaced VCMA and VRCA; some of the more interesting results are presented here.

6.1 Data Sets

Two source data sets, each of length 160,000 samples, were duplicated as many times as necessary at the input of the simulation to provide for the required number of iterations. The first data set was generated by the shell mapping algorithm using a 192-point constellation (Figure 6.1) divided into six rings. Each mapping operation used 36 bits to address one of 2^{36} points in 16-dimensional space, generating an output of eight complex symbols. This data is quite highly shaped with the marginal distributions having a kurtosis κ_s of about 2.64 using the definition

$$\kappa_x = \frac{E|x|^4}{(E|x|^2)^2} \quad (6.1)$$

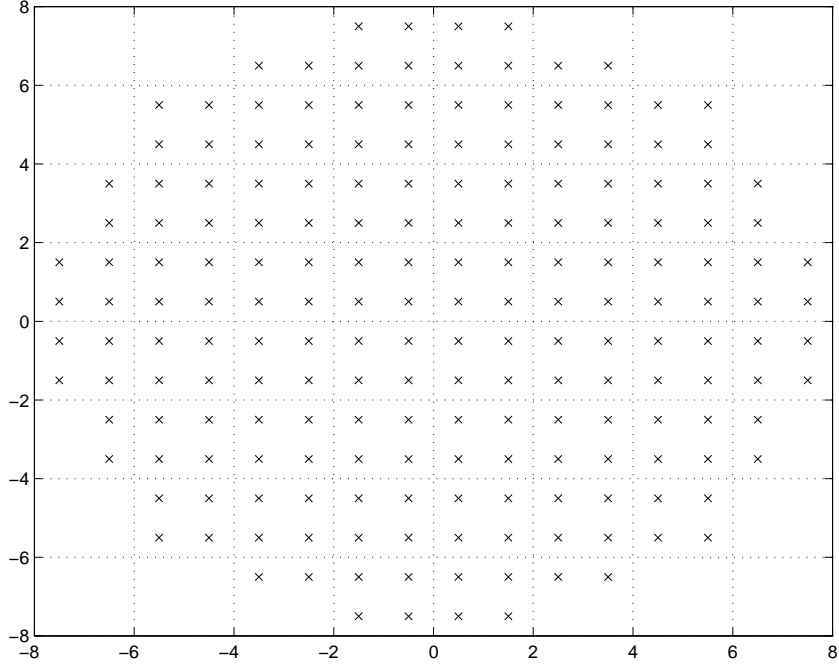


Figure 6.1 192-point constellation used for shell mapping.

where x is a real random variable (other definitions of kurtosis are in common use). Under this definition, a Gaussian distribution has $\kappa = 3$, a uniform distribution has $\kappa = 1.8$, and sub-Gaussian distributions have kurtoses between 1 and 3.

The second data set was generated by a simple trellis shaping implementation (Forney’s “sign bit shaping” [8]) using the convolutional code $[1 + D^2, 1 + D + D^2]$ and 256-QAM (Figure 6.2) for the base constellation. The Viterbi algorithm memory was allowed to span the entire 160,000 samples. The marginal distributions of this data have a kurtosis of about $\kappa_t = 2.49$ using the definition above. This data is therefore less highly shaped than the shell-mapped data.

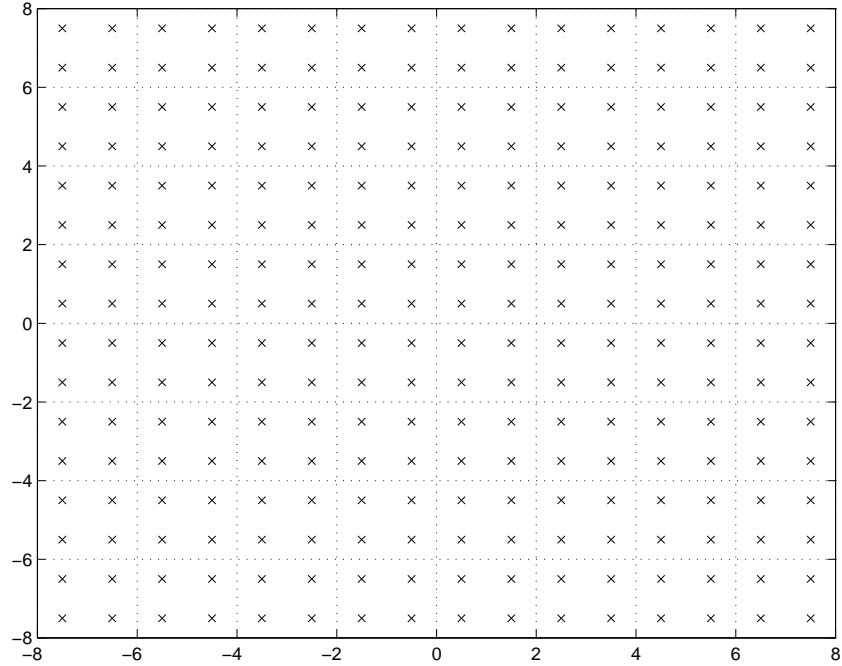


Figure 6.2 256-point constellation used for trellis shaping.

6.2 Channels

A variety of $T/2$ -spaced channels were used in the simulations, but the specific cases presented in this chapter used the following two channels:

$$\mathbf{h}_1 = [0.23, -0.51, 0.84, 0.71, 0, -0.35, 0.61, -0.40] \quad (6.2)$$

$$\mathbf{h}_2 = [0.5, 0.05, 0.1 + j0.4, -0.4, 0, 0.2 - j0.2] \quad (6.3)$$

Note that the second channel will impose a phase shift (constellation rotation) in addition to intersymbol interference.

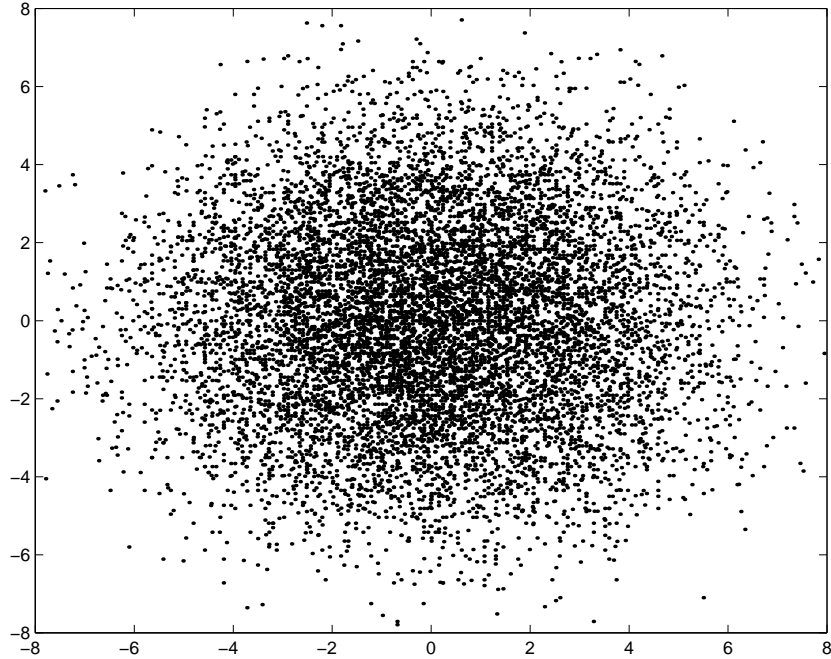


Figure 6.3 Shell-mapped data after passage through channel \mathbf{h}_1 .

6.3 FS-VCMA

Figure 6.3 is a scatter plot showing 10,000 received symbols of the shell-mapped data after passing through channel \mathbf{h}_1 . $T/2$ -spaced VCMA with a step size of $\lambda = 10^{-6}$ and an equalizer of 8 $T/2$ -spaced taps (the same as the delay spread of the channel) was able to equalize this data. Figure 6.4 shows some of the equalized symbols after 80,000 iterations. Figure 6.5 is a plot of the equalizer taps versus time for this simulation.

6.3.1 Improvement of fractionally spaced over baud-spaced VCMA

Figure 6.6 illustrates the result of applying conventional baud-spaced VCMA to the same shell-mapped data and channel as just described. The data was first passed through the $T/2$ -spaced channel, then decimated to T -spaced samples prior to equalization. Six

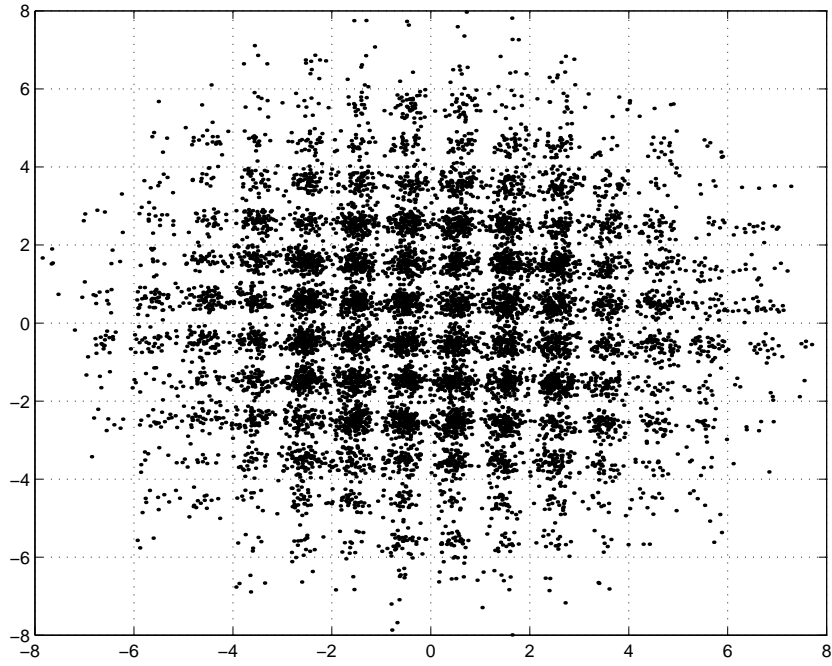


Figure 6.4 FS-VCMA equalizer output for input data in Figure 6.3.

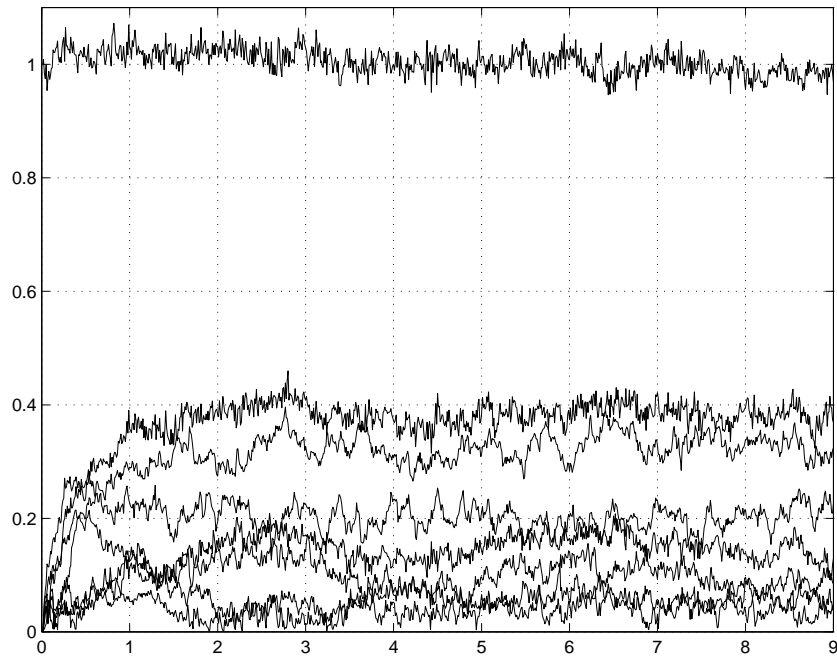


Figure 6.5 FS-VCMA equalizer tap magnitudes vs. number of iterations ($\times 10^4$).

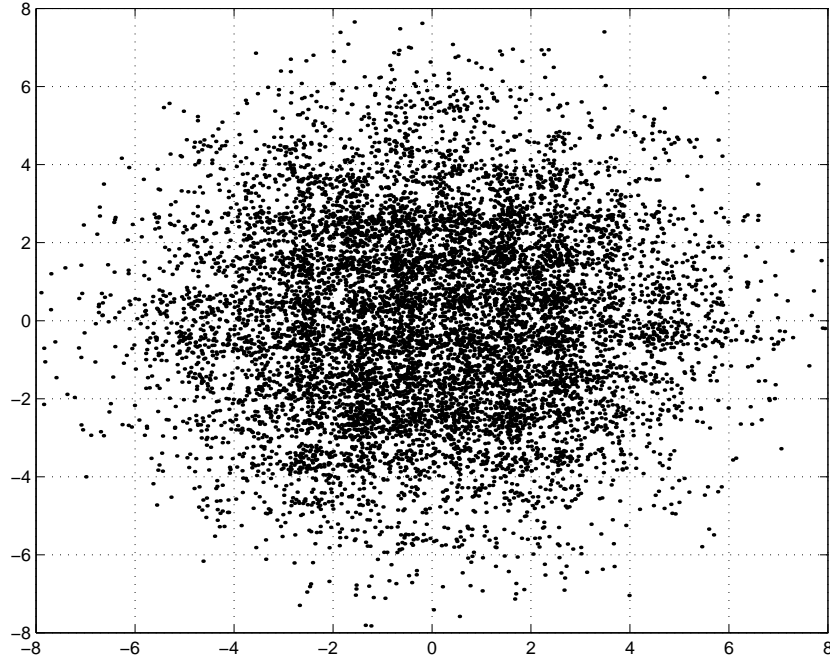


Figure 6.6 Baud-spaced VCMA equalizer output for input data in Figure 6.3.

hundred thousand iterations with a 20-tap equalizer were required to achieve the poor equalization shown here, demonstrating the ability of fractionally spaced VCMA to obtain good equalization with many fewer taps than baud-spaced VCMA, at least over some channels.

6.3.2 Trellis-shaped data

Figure 6.7 is a plot of equalized data points after 60,000 iterations of $T/2$ -spaced VCMA on the trellis-mapped data through channel \mathbf{h}_1 . The step size was $\lambda = 10^{-6}$, and the block size N was set to 8 (the same as for the shell-mapped data). This clearly shows the ability of VCMA to equalize data from shaped sources that did not use shell mapping

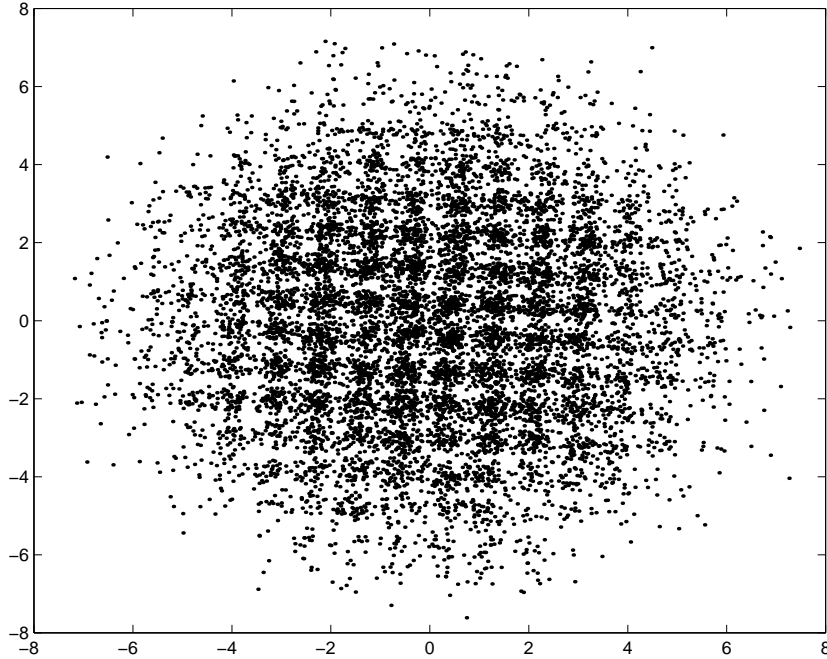


Figure 6.7 Equalized trellis-shaped data through channel \mathbf{h}_1 using FS-VCMA.

as the shaping method; this is in spite of the conceptual link between shell mapping and VCMA discussed in Chapter 2.

6.3.3 Block size N

When equalizing shell-mapped data, the correct value of the block size N is obvious from the shell-mapping algorithm. When other shaping techniques are employed, the appropriate block size is not as clear. In either case, it is natural to ask whether smaller values for N might work, and if so, how well.

Many simulations were performed using channel \mathbf{h}_1 and $T/2$ -spaced VCMA with both shell-mapped and trellis-shaped data to determine the effect of the block size N on the number of iterations required for acceptable equalization. Acceptable equalization was

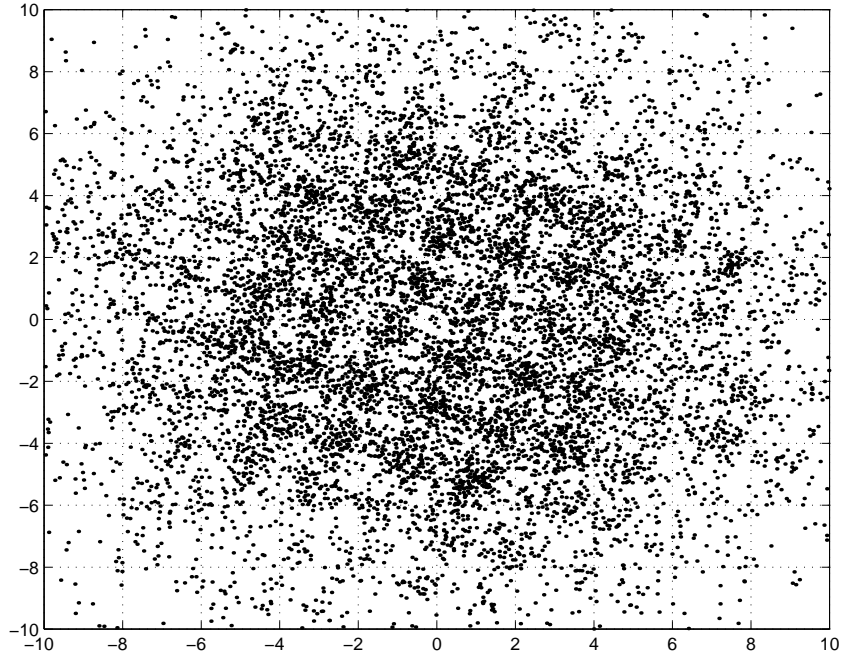


Figure 6.8 Poor equalization of shell-mapped data with FS-VCMA when $N = 2$.

determined by inspection of a scatter plot of the received points. The results are shown in Table 6.1, from which it is clear that N is quite a flexible parameter. Figure 6.8 is a typical result obtained using VCMA when N is too small.

Note that the trellis-shaped data is sufficiently sub-Gaussian that all of the block sizes tested were acceptable. Comparison tests using $T/2$ -spaced CMA either failed completely or had a tendency to drift in and out of good equalization. It should also be noted that these results are just one example; similar results would not necessarily be obtained if a different channel were used.

Table 6.1 Required number of iterations of FS-VCMA (in thousands) vs. block size N .

Block length $N =$	8	7	6	5	4	3	2
Shell mapped data	20	30	30	40	40	50	≈ 200 (poor convergence)
Trellis shaped data	25	30	30	30	40	40	40

6.3.4 Block offset

In many practical situations using shell mapping, the receiver may not know the frame boundaries where one block of N symbols ends and another begins. It would therefore be very desirable if VCMA could operate with arbitrary synchronization to the incoming data. Simulations were performed over channel \mathbf{h}_2 with the shell-mapped data having a block size of 8 complex symbols and with N set to 8 in the equalizer as well. In every case, $T/2$ -spaced VCMA achieved equalization regardless of the block offset used.

6.4 FS-VRCA

Figure 6.9 shows the equalized data points resulting from a $T/2$ -spaced vector reduced constellation algorithm (VRCA) trial of 480,000 iterations with step size $\lambda = 10^{-5}$ and block size $N = 8$ using the trellis-shaped data set. Comparing this with Figure 6.10, an identical simulation using FS-VCMA, we see that the rotation-sensitive nature of the VRCA cost function allowed the algorithm to correct for the constellation rotation which VCMA ignores.

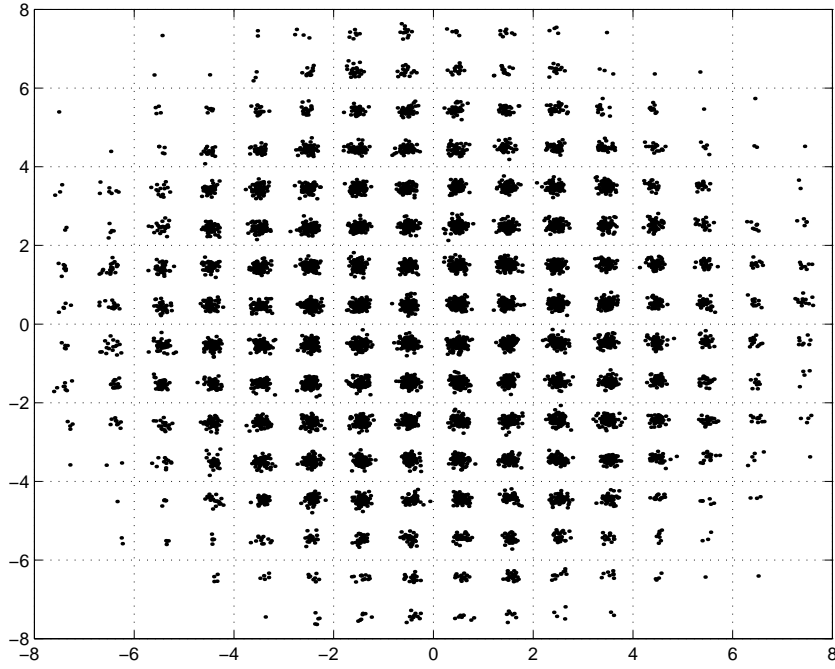


Figure 6.9 Equalized trellis-shaped data through channel h_2 using FS-VRCA.

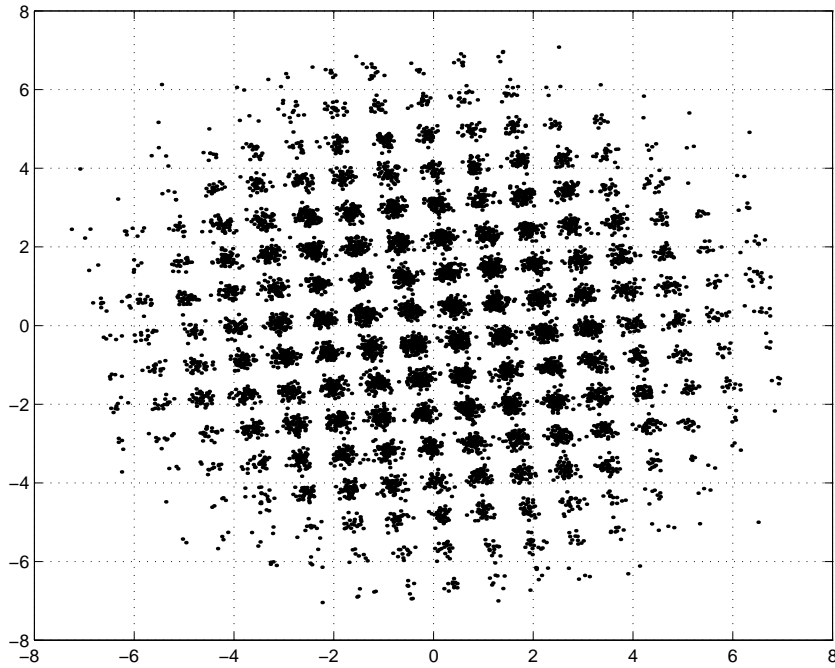


Figure 6.10 Equalized trellis-shaped data through channel h_2 using FS-VCMA.

CHAPTER 7

CONCLUSIONS

The vector constant modulus algorithm, first developed in a baud-spaced equalizer framework with shell-mapped source data, has been shown effective in other scenarios as well. Its ability to work with fractionally spaced equalizers and its modest computational cost over CMA make it viable for use in the increasing number of data communication applications which use source shaping. Indeed, the vector modulus algorithms continue to be the only practical techniques available for the blind equalization of shaped data.

The success of VCMA with both shell-mapped and trellis-shaped data is encouraging. The key concept of using a vector modulus in order to present uniformly-distributed data to the underlying algorithm seems to be valid even when the source data is not generated from a uniform constellation of a particular finite dimension. This suggests that VCMA may remain a viable algorithm for equalizing shaped data regardless of the specific shaping techniques used in the future.

This concept is sufficiently general to be applied to other algorithms based on constant modulus criteria, including the reduced constellation algorithm (RCA) and the multimodulus algorithm (MMA). This yields the vector modulus algorithms VRCA and VMMA. Both algorithms perform about as well as VCMA, but VRCA may be more

applicable when the signal constellation has enough asymmetry to permit automatic correction of a carrier phase offset, thus avoiding the added complexity of a de-rotator block after the equalizer.

Future study into the convergence properties of VCMA and related algorithms would still be desirable, since no proof yet exists showing the global convergence of VCMA. Simulations, however, suggest that it shares similar convergence properties with the well-studied CMA. Further simulations would also be useful to study the performance of VCMA when used with multicarrier signals like OFDM and DMT, which, like single-carrier shaped modulation, have almost Gaussian marginal distributions but decidedly sub-Gaussian vector block distributions.

REFERENCES

- [1] D.N. Godard, “Self-recovering equalization and carrier tracking in two-dimensional data communication systems,” *IEEE Transactions on Communications*, vol. 28, no. 11, pp. 1867–1875, November 1980.
- [2] J.R. Treichler and B.G. Agee, “A new approach to multipath correction of constant modulus signals,” *IEEE Transactions on Communications*, vol. 31, no. 2, pp. 459–472, April 1983.
- [3] C.K. Chan and J.J. Shynk, “Stationary points of the constant modulus algorithm for real Gaussian signals,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 12, pp. 2176–2181, December 1990.
- [4] T. Endres, *Equalizing with fractionally-spaced constant modulus and second-order-statistics blind receivers*, Ph.D. thesis, Cornell University, Ithaca, New York, May 1997.
- [5] V.Y. Yang and D.L. Jones, “A vector constant modulus algorithm for shaped constellation equalization,” *IEEE Signal Processing Letters*, vol. 5, no. 4, pp. 89–91, April 1998.
- [6] M.V. Eyuboglu, G.D. Forney, Jr., P. Dong, and G. Long, “Advanced modulation techniques for V.Fast,” *European Transactions on Telecommunications*, vol. 4, no. 3, pp. 9–22, May–June 1993.

- [7] R. Laroia, N. Farvardin, and S.A. Tretter, “On optimal shaping of multidimensional constellations,” *IEEE Transactions on Information Theory*, vol. 40, no. 4, pp. 1044–1056, July 1994.
- [8] G.D. Forney, Jr., “Trellis shaping,” *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 281–300, March 1992.
- [9] S. Haykin, *Adaptive filter theory*, Prentice Hall, Englewood Cliffs, NJ, 1991.
- [10] V.Y. Yang, “A vector constant modulus algorithm for shaped constellation equalization,” M.S. thesis, University of Illinois at Urbana-Champaign, 1997.
- [11] Y. Li and Z. Ding, “Global convergence of fractionally spaced Godard (CMA) adaptive equalizers,” *IEEE Transactions on Signal Processing*, vol. 44, no. 4, pp. 818–826, April 1996.
- [12] G.J. Foschini, “Equalizing without altering or detecting data,” *AT&T Technical Journal*, vol. 64, pp. 1885–1911, October 1985.
- [13] Z. Ding, R. Johnson, Jr., and R. Kennedy, “On the (non)existence of undesirable equilibria of Godard blind equalizers,” *IEEE Transactions on Signal Processing*, vol. 40, no. 10, pp. 2425–2432, October 1992.
- [14] O. Shalvi and E. Weinstein, “New criteria for blind deconvolution of nonminimum phase systems (channels),” *IEEE Transactions on Information Theory*, vol. 36, pp. 312–321, March 1990.
- [15] J.F. Claerbout, *Fundamentals of geophysical data processing, with applications to petroleum prospecting*, McGraw-Hill, New York, 1976.

- [16] C.B. Papadias, “On the existence of undesirable global minima of Godard equalizers,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Munich, Germany, April 1997, pp. 3941–3944.
- [17] D.N. Godard and P.E. Thirion, “Method and device for training an adaptive equalizer by means of an unknown data signal in a QAM transmission system,” U.S. Pat. 4 227 152, October 1980.
- [18] L.M. Garth, J. Yang, and J. Werner, “Blind start-up for VDSL systems,” Unpublished manuscript.
- [19] J. Yang, J.J. Werner, and G.A. Dumont, “The multimodulus blind equalization algorithm,” in *Proc. Thirteenth Intl. Conf. on Digital Signal Processing*, Santorini, Greece, July 1997, pp. 127–130.