

# THE FRACTIONALLY SPACED VECTOR CONSTANT MODULUS ALGORITHM

Mark A. Haun and Douglas L. Jones

Department of Electrical and Computer Engineering  
University of Illinois at Urbana-Champaign  
markhaun@uiuc.edu, dl-jones@uiuc.edu

## 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. Some simple changes in the structure of VCMA allow it to be used in fractionally-spaced equalizers with their attendant benefits. Although developed with shell mapping in mind, VCMA can also equalize data from other shaping methods such as trellis shaping. Furthermore, the vector modulus concept from VCMA can be successfully applied to other algorithms based on constant modulus criteria, including RCA and MMA. Simulations have verified all of these results.

## 1. INTRODUCTION

The constant modulus algorithm (CMA), first described in [1] and [2], is a popular algorithm for blind equalization in digital communication systems. The convergence of CMA, however, slows down and eventually fails as the source data approaches a Gaussian distribution [3]. Unfortunately, an approximately Gaussian source is precisely the goal of source shaping techniques which increase system performance using non-equiprobable signaling.

Source shaping is able to provide gain, independent of coding, by using efficient constellation shapes. In two dimensions, a circular constellation has an inherent shaping gain of about 0.2 dB over the more traditional square. 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  $N \rightarrow \infty$ . In the limit, this is equivalent to imposing a Gaussian distribution on the two-dimensional signal points.

As shaped signal constellations become more common, an alternative blind equalization algorithm is clearly needed. The vector constant modulus algorithm (VCMA) [4] was re-

cently introduced as an extension of CMA which can equalize data from sources having nearly Gaussian marginal distributions. Source shaping is assumed to be 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 later, this is in fact how the shaping is done.) Vectors of  $N$  successive samples should 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.

## 2. THE VECTOR CONSTANT MODULUS ALGORITHM

Consider a complex baseband model of a digital communication system. 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  $\mathbf{z} = \mathbf{y} * \mathbf{c}$  where  $\mathbf{y} = \mathbf{a} * \mathbf{h}$  is the channel output.

$\mathbf{a}_n = [a_n \ a_{n-1} \ \cdots \ a_{n-N+1}]$  is a vector of  $N$  transmitted complex signal points corresponding to a point in the  $2N$ -dimensional uniform constellation.

$\mathbf{z}_n = [z_n \ z_{n-1} \ \cdots \ z_{n-N+1}]$  is a vector of  $N$  output samples from the equalizer, and is calculated by  $\mathbf{z}_n = \mathbf{Y}_n' \mathbf{c}_n$ , where

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

and  $L_c$  is the number of equalizer taps. Note there 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:

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

This work 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.

where

$$R_p = \frac{E|\mathbf{a}_n|^{2p}}{E|\mathbf{a}_n|^p} \quad (3)$$

is a constant dependent on the signal constellation, which acts as a scaling factor for the equalized data.

A stochastic gradient update with step size  $\mu$  is used to update the filter coefficients.

$$\mathbf{c}_{n+1} = \mathbf{c}_n - \mu \left[ \frac{\partial C F_{\text{VCMA}}}{\partial \mathbf{c}_n} \right] \quad (4)$$

The derivative is calculated as

$$\left[ \frac{\partial C F_{\text{VCMA}}}{\partial \mathbf{c}_n} \right] = E \left[ 2p \mathbf{Y}_n^* (\mathbf{Y}'_n \mathbf{c}_n) |\mathbf{Y}'_n \mathbf{c}_n|^{p-2} (|\mathbf{Y}'_n \mathbf{c}_n|^p - R_p) \right] \quad (5)$$

using the result

$$\frac{\partial}{\partial \mathbf{c}_n} |\mathbf{Y}'_n \mathbf{c}_n| = \mathbf{Y}_n^* (\mathbf{Y}'_n \mathbf{c}_n) |\mathbf{Y}'_n \mathbf{c}_n|^{-1} \quad (6)$$

Finally, the expectation is dropped to obtain the desired tap update equation

$$\mathbf{c}_{n+1} = \mathbf{c}_n - \lambda \mathbf{Y}_n^* \mathbf{z}_n |\mathbf{z}_n|^{p-2} (|\mathbf{z}_n|^p - R_p) \quad (7)$$

with step size  $\lambda$ .  $p$  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 shortcuts are possible, most significantly the recursive update of the product  $\mathbf{Y}_n^* \mathbf{z}_n$ . The end result is a cost of  $14L_c + 5$  real multiplications and  $14L_c + 3$  real additions per symbol period, which is only about 50% more expensive than conventional CMA.

### 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 filtering operations such as matched filtering and adjustment of sampling phase. 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 [5]. 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  $\mathbf{z}_n = \mathbf{Y}'_n \mathbf{c}_n$ , with the rectangular matrix  $\mathbf{Y}_n$  given by (1) 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

$$\mathbf{Y}_n = \begin{bmatrix} y_n & y_{n-2} & \cdots & y_{n-2N+2} \\ y_{n-1} & y_{n-3} & \cdots & y_{n-2N+1} \\ \vdots & \vdots & & \vdots \\ y_{n-L_c+1} & y_{n-L_c-1} & \cdots & y_{n-L_c-2N+3} \end{bmatrix} \quad (8)$$

Simulations performed with FSE VCMA have verified its ability to equalize several short FIR channels with a required equalizer length comparable to the channel delay spread.

### 4. APPLICABILITY TO DIFFERENT SOURCE SHAPING METHODS

Shell mapping is the most common source shaping technique in current use and is the shaping method chosen for the latest telephone modem standards [6]. In shell mapping, a two-dimensional circular constellation is divided into  $M$  rings of increasing energy, with each ring having an equal number of points and a cost label proportional to the energy of the points. Now consider the  $2N$ -dimensional set of points described by combinations of  $N$  points in 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. The cost is determined by adding the  $N$  individual ring costs, and the sub-constellation selected by shell mapping is nearly a sphere in  $2N$  dimensions. Then,  $b$  bits can be sent using  $N$  of the two-dimensional signal points transmitted in sequence.

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 a uniform distribution of signal vectors and get around the problem of the almost-Gaussian distribution of signal points in two dimensions [4]. 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.

A different shaping approach, trellis shaping, was de-

scribed by Forney in [7]. 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. It is not possible to describe this operation in a specific, finite-dimensional space, but 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.

## 5. GENERALIZATION TO RELATED ALGORITHMS

The vector modulus concept from VCMA can be applied to other blind equalization algorithms having cost functions similar to CMA. For example, the reduced constellation algorithm (RCA) proposed in [8] uses the cost function

$$CF_{RCA} = E|z_n - Rcsgn(z_n)|^2 \quad (9)$$

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 [9], 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. A vector RCA (VRCA) is easily derived by using vector quantities in place of the scalar quantities.

$$CF_{VRCA} = E|\mathbf{z}_n - Rcsgn(\mathbf{z}_n)|^2 \quad (10)$$

The complex sign function  $csgn$  returns a vector the same length as  $\mathbf{z}$  on an element-by-element basis. VRCA will therefore attempt to fit vectors of received symbols to the corners of a hypercube. The update equation for VRCA is

$$\mathbf{c}_{n+1} = \mathbf{c}_n - \lambda \mathbf{Y}_n^* (\mathbf{z}_n - Rcsgn(\mathbf{z}_n)) \quad (11)$$

Computationally, VRCA is very similar to 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.

A similar derivation and experiments have been carried out with a vector version of the multimodulus algorithm (MMA) described in [10]. Simulations yielded results much like those for VRCA, and similar comments apply.

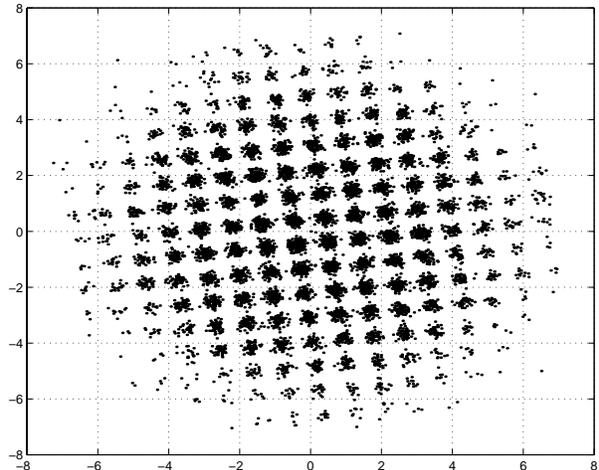


Figure 1: Equalized trellis-shaped data using VCMA

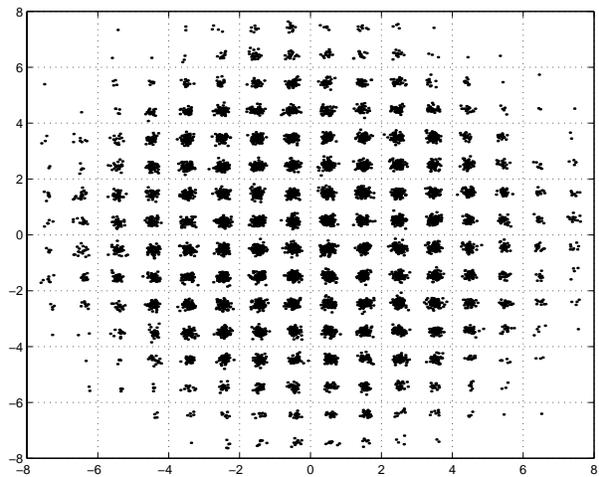


Figure 2: Equalized trellis-shaped data using VRCA

## 6. SIMULATIONS

An extensive set of simulations was performed using fractionally-spaced VCMA and VRCA, and a few of the results are shown in the figures. In all of the cases presented here, the length of the equalizer  $L_c$  was eight taps, the input vector length  $N$  was eight, and a  $T/2$ -spaced FIR system

$$\mathbf{h} = [0.5, 0.05, 0.1 + j0.4, -0.4, 0, 0.2 - j0.2] \quad (12)$$

was used as the channel model. 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 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 a kurtosis  $\kappa$  of 1.75 using the definition

$$\kappa_x = \frac{E|x|^4}{(E|x|^2)^2} \quad (13)$$

where  $x$  is a complex random variable (other definitions of kurtosis are in common use). Under this definition, a complex Gaussian source has  $\kappa = 2$ , a uniformly distributed source has  $\kappa = 1.4$ , and sub-Gaussian sources have kurtoses between these two extremes. A real-valued Gaussian source has  $\kappa = 3$ , a somewhat more familiar result.

The second data set was generated by a simple trellis shaping implementation using the convolutional code  $[1 + D^2, 1 + D + D^2]$  and 256-QAM for the base constellation. The Viterbi algorithm memory was allowed to span the entire 160,000 samples. This data has kurtosis  $\kappa = 1.64$  using the definition above and is less highly shaped than the shell-mapped data.

Figure 1 is a plot of equalized data points near the end of a 480,000 iteration run of  $T/2$ -spaced VCMA on trellis-mapped data through the channel described above. The step size was  $\lambda = 10^{-6}$ . As expected, an equalizer of length comparable to the channel is able to equalize the data. Although an input data vector length  $N$  of 8 was used for this trial,  $N$  values as low as 3 or 4 still allowed successful convergence. Similar results were obtained with an identical simulation on shell-mapped data. Comparison trials using  $T/2$ -spaced CMA either failed completely or had a tendency to drift in and out of good equalization.

Figure 2 shows the equalized data points resulting from a vector reduced constellation algorithm (VRCA) trial of 480,000 iterations with step size  $\lambda = 10^{-5}$  using the trellis-shaped data set. Comparing this with Figure 1, we see that the rotation-sensitive nature of the VRCA cost function allowed the algorithm to correct for the constellation rotation which VCMA ignores.

## 7. CONCLUSION

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. The success of VCMA with both shell-mapped and trellis-shaped data suggests that it will work with data produced by other shaping methods as well. 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 concept is sufficiently general to be applied to other algorithms based on constant modulus criteria.

Additionally, following the traditional convergence analysis for CMA, a proof has now been developed showing VCMA to be globally convergent to a perfectly equalizing setting when the equalizer is infinitely parameterized; this can be found in [11].

## 8. 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, Nov. 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, Dec. 1990.
- [4] 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.
- [5] 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.
- [6] M. V. Eyuboglu, G. D. Forney Jr., P. Dong and G. Long, "Advanced modulation techniques for VFast," *European Transactions on Telecommunications*, vol. 4, no. 3, pp. 9-22, May-June 1993.
- [7] G. D. Forney, Jr., "Trellis shaping," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 281-300, March 1992.
- [8] 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, Oct. 7, 1980.
- [9] L. M. Garth, J. Yang, and J. Werner, "Blind start-up for VDSL systems," unpublished manuscript.
- [10] J. Yang, J. J. Werner and G. A. Dumont, "The multimodulus blind equalization algorithm," *Proc. Thirteenth Intl. Conf. on Digital Signal Processing*, Santorini, Greece, July 2-4, 1997.
- [11] M. Haun, M.S. thesis, University of Illinois at Urbana-Champaign, Urbana, IL, Dec. 1998.