An Eight-parallel Hierarchical Constant Modulus Algorithm for Sixty-four Quadrature Amplitude Modulation

Chao GAO\textsuperscript{1,2,3}, Xiaojie WU\textsuperscript{3}, Libin JIAO\textsuperscript{1,2} and Xiaofeng LI\textsuperscript{3,*}

\textsuperscript{1} The 54th Research Institute of China Electronics Technology Group Corporation, 589 West Zhongshan Road, Shijiazhuang, China
\textsuperscript{2} Science and Technology on Information Transmission and Dissemination in Communication Networks Laboratory, 589 West Zhongshan Road, Shijiazhuang, China
\textsuperscript{3} School of Astronautics and Aeronautic, University of Electronic Science and Technology of China, 2006 Xiyuan Avenue, Chengdu, China

E-mail: lxf3203433@uestc.edu.cn

Abstract. One derivative type of the constant modulus algorithm (CMA) termed hierarchical CMA (HCMA) is investigated for its parallel implementation with 8 lanes. Taking the essence of CMA, HCMA is also the steepest descent under the least mean-square (LMS) criterion theoretically. To update the tap coefficients of the equalizer, HCMA utilizes adjustable modulus in accordance with the dynamical amplitude of the baseband signals. The above mechanism of HCMA can make the expectation of the error estimation converge to 0. The 8-parallel HCMA adopts the framework of the pipelined LMS adaptive filter. On the basis of the iterated short convolution and the relaxed look-ahead approximation, the implementation of 8-parallel HCMA can reduce much multiplication. The numerical simulation indicates that HCMA performs better than CMA with evident gain for the equalization of 64 quadrature amplitude modulation (QAM).

1. Introduction

The adaptive equalization technique is definitely an indispensable branch in the field of the wireless communication. The equalizer in the demodulator of the receiver can eliminate the inter-symbol interference caused by the multipath fading channel. For practical application, the priori knowledge about the time-variant channel response always tends to be unavailable. To deal with this situation with unbeknownst channel, various blind equalizers have been investigated in the past few decades [1].

As one famous linear blind equalizer, CMA has drawn much attention for its concise mathematical expressions, low computational complexity and efficient serial implementation. CMA is the steepest descent under the LMS criterion. Additionally, independent of carrier phase, CMA can equalize the distorted baseband signals before the carrier synchronization in the demodulator [2].

Nevertheless, CMA still encounters several challenges [3, 4]. On one hand, in the situation of high-speed communication, the symbol rate of the baseband signal may be so high that exceeds the valid clock frequency of the field programmable gate array (FPGA) device. Thus, it is
necessary to modify CMA under the parallel architecture and simultaneously economize the multiplier resource in FPGA such as DSP48 slices. On the other hand, CMA is inappropriate to equalize those high-order QAM baseband signal. The reason is obvious. Take 64 QAM into account. The constellation of 64 QAM with \( \{ I + Q i | I, Q \in \{ \pm 1, \pm 3, \pm 5, \pm 7 \} \} \) has multiple values of the modulus \( \{ \sqrt{2}, \sqrt{10}, \sqrt{18}, \sqrt{26}, \sqrt{34}, \sqrt{50}, \sqrt{58}, \sqrt{74}, \sqrt{98} \} \), depicted as concentric circles in figure 1. Consequently the expectation of the error estimation from CMA fails to converge to 0. To solve the problem, HCMA is one feasible algorithm, which not only takes the essence of CMA but also relieve certain restriction of CMA.

![Figure 1. Constellation of 64 QAM with different values of the modulus.](image)

Regarded as increment to the previous published literature, this paper mainly focuses on the 8-parallel HCMA for 64 QAM. The rest of the paper is organized as follows. Section 2 presents the derivation of HCMA. In section 3, simulation is performed, followed by conclusion in section 4.

2. Derivation
The serial implementation of CMA can be described as [1]

\[
y(n) = W^T(n) X(n),
\]

\[
e(n) = y(n) \left( R_C^2 - |y(n)|^2 \right),
\]

\[
W(n+1) = W(n) + \mu e(n) X^C(n),
\]

where \( X(n) = (x(n), \cdots, x(n-N+1))^T \) is the input sequence, \( y(n) \) is the output sequence, \( W(n) = (w_0(n), \cdots, w_{N-1}(n))^T \) is the tap coefficients, and \( e(n) \) is the error sequence. \( R_C \) is the average modulus of \( x(n) \), and \( R_C^2 = 58 \) for 64 QAM in figure 1. \( \mu \) is the step-size parameter.
which ensures the convergence of the iterative procedure. The superscripts $^T$ and $^C$ stand for transposition and conjugation, respectively.

To parallel CMA, equations (1) and (3) will be modified to the pipelined LMS-type adaptive filter [5]. This paper adopts 8 parallel lanes, which can support high-speed communication with 9.6 Gbps when the clock frequency of FPGA is 200 MHz for 64 QAM. For mathematical convenience, $N = 32$, which can be divided exactly by 8.

Based on the iterated short convolution, the parallel implementation of equation (1) can be described as [6]

$$Y_8 = (P^T_2 \otimes P^T_4) H_8 (Q^T_2 \otimes Q^T_4) A^T_8 X_8,$$  \hspace{1cm} (4)

where

$$X_8 = \begin{pmatrix}
\sum_{n=0}^{k} z^{-nD} x (8 (k - n) + 7) \\
\vdots \\
\sum_{n=0}^{k} z^{-nD} x (8 (k - n) - 7)
\end{pmatrix},$$  \hspace{1cm} (5)

and

$$Y_8 = \begin{pmatrix}
\sum_{n=0}^{k} z^{-nD} y (8 (k - n) + 7) \\
\vdots \\
\sum_{n=0}^{k} z^{-nD} y (8 (k - n) + 0)
\end{pmatrix}.$$  \hspace{1cm} (6)

$z^{-nD}$ stands for delaying $n$ clock periods. The preaddition matrices include

$$P^T_2 = \begin{pmatrix}
1 & 1 & 0 \\
0 & -1 & 1
\end{pmatrix},$$  \hspace{1cm} (7)

and

$$P^T_4 = \begin{pmatrix}
1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1
\end{pmatrix},$$  \hspace{1cm} (8)

whereas the postaddition matrices include

$$Q_2 = \begin{pmatrix}
1 & 0 & 0 \\
1 & -1 & 1
\end{pmatrix},$$  \hspace{1cm} (9)

and

$$Q_4 = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
-1 & 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 \\
0 & -1 & 1 & -1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & -1 & -1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}.$$  \hspace{1cm} (10)
The symbol “⊗” stands for the Kronecker tensor product. Other arguments in equation (4) include

\[ H_8 = \text{diag} \begin{pmatrix} \frac{3}{n=0} z^{-nD} w_{8n} (8k) \\ \vdots \\ \frac{3}{n=0} z^{-nD} w_{8n+7} (8k) \end{pmatrix} \]  

(11)

and

\[ A_8 = \left( (I_7, O_{7 \times 8})^T, (O_{7 \times 4}, I_7, O_{7 \times 4})^T, (O_{7 \times 8}, I_7)^T \right), \]  

(12)

where the symbols “\( I_m \)” and “\( O_{m \times n} \)” stand for the \( m \)-by-\( m \) identity matrix and the \( m \)-by-\( n \) zeros matrix, respectively.

The parallel implementation of equation (3) can be described as

\[ W(8(k+1)) = W(8(k+1)) + \mu \frac{7}{n=0} e(8k+n) X^C(8k+n), \]  

(13)

which will occupy huge multiplier resource when implementing in FPGA. Based on the relaxed look-ahead approximation, equation (13) becomes

\[ W(8(k+1)) \approx W(8(k+1)) + \mu \frac{1}{n=0} e(8k+n) X^C(8k+n), \]  

(14)

which has lower complexity [7].

**Figure 2.** Relationship between \( R_H^2 \) and \( |y(n)| \) for HCMA.

To deduce HCMA from CMA, equation (2) should be generalized to

\[ e(n) = y(n) \left( R_H^2 (|y(n)|) - |y(n)|^2 \right). \]  

(15)
Figure 2 depicts the relationship between $R_H^2$ and $|y(n)|$ for HCMA. The parallel implementation of equation (15) is to multiplex itself. It is apparent that equations (2) and (15) have analogical expression to each other. Thus, the parallel architectures of CMA and HCMA are equivalent. Since HCMA can hardly work after cold boot, the demodulator has to first implement CMA for initialization and then switch to HCMA. Their equivalent architectures make the switch seamless.

3. Simulation
This section adopts the mean square error

$$MSE(k) = (1 - \theta) \cdot MSE(k - 1) + \theta \cdot \frac{1}{2} \sum_{n=0}^{1} (R_H(|y(8k + n)|) - |y(8k + n)|)^2$$

(16)

to evaluate the feasibility and applicability of 8-parallel HCMA, where $\theta$ is the update factor. Equation (16) is convenient to be transplanted into FPGA.

![Figure 3. MSE of CMA and HCMA.](image1)

![Figure 4. Outputs of HCMA.](image2)

The parameters relevant to the simulation include: $\mu = 2^{-22}$, $\theta = 2^{-7}$, $Eb/N0 = 25$ dB for input signal, channel response $[0.0410 + 0.0109j, 0.0495 + 0.0123j, 0.0672 + 0.0170j, 0.0919 + 0.0235j, 0.7920 + 0.1281j, 0.3960 + 0.0871j, 0.2715 + 0.0498j, 0.2291 + 0.0414j, 0.1287 + 0.0154j, 0.1032 + 0.0119j]$.

Figure 3 depicts the mean square error of CMA and HCMA during the iteration, respectively. It must be pointed that HCMA is inactivated until $k \geq 2 \times 10^4$. According to figure 3, HCMA performs better than CMA with 2 dB gain. Figure 4 depicts the last $1 \times 10^4$ outputs of HCMA. The output signal gathers to the constellation of 64 QAM with rotation by certain fixed angle, which conforms to expectation.

4. Conclusion
This paper mainly discusses HCMA under the parallel architecture with 8 lanes. HCMA inherits those representative properties of CMA to equalize the deteriorative baseband signals after transmission through the multipath fading channel. Additionally, HCMA utilizes alterable
multiple moduli rather than single constant modulus to update the tap coefficients of the equalizer. Hence, HCMA can be regarded as one generalized upgrade of CMA. It must be pointed out that when demodulating the baseband signals, HCMA could hardly work regularly unless CMA was conducted for initialization. The serial implementation of HCMA can be summarized by 3 formulae, which correspond to the outputs, the errors and the tap coefficients in order. The parallel implementation of HCMA is to modify these formulae by the pipelined LMS-type adaptive filter. The iterated short convolution and the relaxed look-ahead approximation are utilized to reduce the potential huge multiplication in computing the outputs and updating the tap coefficients, respectively. To validate the feasibility and applicability of the 8-parallel HCMA, a stochastic sequence of 64 QAM with noise is generated. The numerical simulation indicates that HCMA is able to achieve higher gain than CMA.

References
[1] Proakis J and Salehi M 2007 Digital Communications 5th ed (New York: McGraw-Hill Higher Education)
[2] Godard D N 1980 IEEE Transactions on Communications 28 1867-1875
[3] Lin C, Zhang J and Shao B 2012 IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E95.A 1412-1415
[4] He L, Amin M G, Reed C and Malkemes R C 2004 IEEE Transactions on Signal Processing 52 2058-2069
[5] Parhi K K 1999 VLSI Digital Signal Processing Systems Design and Implementation (New York: Wiley-Interscience)
[6] Cheng C and Parhi K K 2004 IEEE Transactions on Circuits and Systems I: Regular Papers 51 1492-1500
[7] Shanbhag N R and Parhi K K 1993 IEEE Transactions on Circuits and Systems II: Express Briefs 40 753-766