CONSTANT MODULUS PERFORMANCE SEARCH USING NEWTON'S METHOD

Tokunbo Ogunfunmi, Senior Member, IEEE

Santa Clara University
Electrical Engineering Department
500 El Camino Real, Santa Clara CA 95053-0569

Hamadi Jamali, Member, IEEE

Canon R&D Center Americas, Inc.
3300 North First Street
San Jose, CA 95134

ABSTRACT

This paper uses Newton's method to seek the global minimum of the constant modulus performance measure. Unlike the common practice of using the constant modulus adaptive algorithm, the new approach does not suffer from local minima. The paper also discusses some implementation issues of the new algorithm.

1. INTRODUCTION

The Constant Modulus (CM) minimization problem can be stated as follows: *Given a set of n samples from a discrete time complex valued random sequence $\{x_t\}$, it is desired to use these samples in order to obtain an estimate $\{y_t\}$ of a correlated, but unobservable, sequence $\{s_t\}$.*

The estimate $\{y_t\}$ is required to be of the form:

$$y_t = W^* X_t \tag{1}$$

where $X_t = [x_t \, x_{t-1} \, \ldots \, x_{t-n+1}]'$ is the $n \times 1$ vector of the sample values, $W = [w_0 \, w_1 \, \ldots \, w_{n-1}]'$ is the $n \times 1$ vector of unknown, but constant parameters, and the superscripts $'$ and * are the real and the complex transpose operators, respectively. The accuracy of this estimator is measured by the CM, or the Godard, error criterion $\mathcal{J}$ given independently by [1] and [2] as:

$$\mathcal{J} = \frac{1}{4} E[(| y_t |^2 - \gamma)^2] \tag{2}$$

where $| y_t |$ is the modulus or the norm of the sequence $\{y_t\}$, the symbol $E[.]$ is the mathematical expectation operator taken over all possible sequences, and the symbol γ is the dispersion constant of the sequence $\{s_t\}$. This performance measure has proven effective with a number of digitally modulated signals including PAM, PSK and QAM.

A major appeal of the CM minimization is that it can be implemented in real time using the Constant Modulus Adaptive (CMA) algorithm given by:

$$W_t = W_{t-1} - \mu \varepsilon_{t-1} y_{t-1} X_{t-1} \tag{3}$$

where μ is a constant step size, W_t is an estimate which, presumably, reaches the optimum coefficient vector W_{opt} as t tends toward infinity, and $\varepsilon_t = | y_t |^2 - \gamma$. Commonly, γ is set to 1.

The CMA algorithm is widely studied in the literature [3]. In particular, it is known that this algorithm can lead to erroneous solutions [4]. This contribution proposes a new algorithm for searching the surface of the CM performance measure that does not exhibit local minima. Specifically, the paper uses Newton's method to obtain a solution in the kronecker product parameter space. The obtained augmented solution is then converted back to yield the filter weights. Our results are expressed in terms of the statistics of the sample sequence alone. As such, they apply to other areas besides equalization. Section 2 derives the new algorithm. Section 3 discusses some implementation issues. Section 4 offers concluding remarks.

2. NEWTON'S METHOD BASED CMA

This section introduces a new iterative method for searching the CM function that does not converge to local minima.

Theorem: Given the $l \times 1$ correlation vector P_φ and the $l \times l$ fourth order moment matrix $R_{\varphi\varphi}$ of the received sequence $\{x_t\}$, and given an arbitrary initial value, $\theta(-1)$, of an unknown but constant $l \times 1$ vector θ, a variation of Newton's method for searching the constant modulus cost function is formulated as:

$$For \ k = 0 \ to \ k = N-1,$$

$$\nabla(k-1) = R_{\varphi\varphi}\theta(k-1) - P_\varphi \tag{4}$$

$$\theta(k) = \theta(k-1) - \mu R_{\varphi\varphi}^{-1} \nabla(k-1) \tag{5}$$

$$w_i(k) = \sqrt{| \tau_{i(n+1)}(k) |} \phi(\tau_{i(n+1)}(k)) \tag{6}$$

$$i = 0, 1, \ldots, n-1$$

where the symbol N is the number of iterations needed for the algorithm to converge to an acceptable solution, the variables τ_i and $\phi(\tau_i)$ are the i^{th} element of the vector θ and its phase, respectively, the parameter w_i is the i^{th} element of the coefficient vector W, and the variable l is obtained from the data length n as $l = n^2$. The elements w_i of the parameter vector W in equation (6) can also be extracted from the elements τ_i of the augmented parameter vector θ as follows:

$$w_i(k) = \frac{\tau_i(k)}{\sqrt{|\ \tau_0(k)\ |}}, \quad \tau_0(k) \neq 0, \\ \forall k, \quad i = 0, 1, \ldots, n-1 \tag{7}$$

For real data, equation (7) remains the same, but equation (6) becomes:

$$w_i(k) = \pm\sqrt{|\ \tau_j(k)\ |}, \quad i = 0, 1, \ldots, n-1 \tag{8}$$

where the index j is given as:

$$j = in - \sum_{k=0}^{i-1} k$$

Proof: Beginning with the equation of the CM cost function in (2), we replace the signal $\{y_t\}$ by its formula from equation (1). Then, we expand the expression of the functional $\mathcal{J}$ to obtain:

$$\begin{aligned} \mathcal{J} &= \frac{1}{4}E[J_0] - \frac{\gamma}{2}E[J_1] + \frac{\gamma^2}{4} \\ J_0 &= (W^* X_t X_t^* W)(W^* X_t X_t^* W) \\ J_1 &= W^* X_t X_t^* W \end{aligned} \tag{9}$$

To proceed, define the $l \times 1$ kronecker product coefficients vector θ and the $l \times 1$ kronecker product data vector φ_t as:

$$\begin{aligned} \theta &= W \otimes \overline{W} = [w_0 W^*\, w_1 W^*\ \ldots\ w_{n-1} W^*]' \\ \varphi_t &= X_t \otimes \overline{X}_t = [x_t X_t^*\, x_{t-1} X_t^*\ \ldots\ x_{t-n+1} X_t^*]' \end{aligned}$$

where $\otimes$ designates the kronecker product operator [5]. Then, notice that the term J_1 can be written as:

$$J_1 = (W \otimes \overline{W})^*(X_t \otimes \overline{X}_t)$$

Similarly, the term J_0 can be expressed as:

$$J_0 = (W \otimes \overline{W})^*(X_t \otimes \overline{X}_t)(X_t \otimes \overline{X}_t)^*(W \otimes \overline{W})$$

Hence, the cost function $\mathcal{J}$ in (9) becomes:

$$\begin{aligned} \mathcal{J} &= \frac{1}{4}\theta^* R_{\varphi\varphi}\theta - \frac{\gamma}{2}\theta^* P_\varphi + \frac{\gamma^2}{4} \\ P_\varphi &= E[\varphi_t] \\ R_{\varphi\varphi} &= E[\varphi_t \varphi_t^*] \end{aligned} \tag{10}$$

Notice that the term $\theta^* R_{\varphi\varphi}\theta$ is given by:

$$\theta^* R_{\varphi\varphi}\theta = E[|\ y_t\ |^4]$$

Since $E[|\ y_t\ |^4] \geq 0$, we conclude that the matrix $R_{\varphi\varphi}$ is Hermitian and positive semi–definite. Thus, the quadratic form in equation (10) is convex. Without loss of generality, we also assume here that the matrix $R_{\varphi\varphi}$ is nonsingular. Consequently, the gradient $\nabla_\theta J$ of the CM cost function $\mathcal{J}$ in equation (10), with respect to the new parameter vector θ, is given by:

$$\nabla_\theta J = \left[\frac{\partial J}{\partial \theta_0}\frac{\partial J}{\partial \theta_1}\ldots\frac{\partial J}{\partial \theta_{l-1}}\right]' \tag{11}$$

where the variable θ_i is the i^{th} component of the vector θ, the quantity $\frac{\partial J}{\partial \theta_i} = \frac{\partial J}{\partial \Re(\theta_i)} + j\frac{\partial J}{\partial \Im(\theta_i)}$ defines the partial derivative of the real scalar function $\mathcal{J}$ with respect to the complex variable θ_i, with $\Re(\theta_i)$ and $\Im(\theta_i)$ being the real and the imaginary parts of the complex variable θ_i. Evaluating each of the terms, the expression (11) becomes:

$$\nabla_\theta J = R_{\varphi\varphi}\theta - P_\varphi \tag{12}$$

Notice that equation (12) is the same as equation (4) where the time index $k-1$ has been omitted from equation (12). This shows that equation (4) is the gradient of the CM performance cost function with respect to the augmented parameter vector θ.

Now, the Hessian matrix $\mathcal{H}_\theta J$ of the second order derivatives of the CM function with respect to the augmented parameter vector θ is computed as:

$$\mathcal{H}_\theta J = \nabla_\theta(\nabla'_\theta J) = R_{\varphi\varphi} \tag{13}$$

Using equations (12), (13) and (14), the innovation term $R_{\varphi\varphi}^{-1}\nabla(k-1)$ in equation (5) can be written as:

$$R_{\varphi\varphi}^{-1}\nabla(k-1) = [\mathcal{H}_\theta J]^{-1}\nabla_\theta J \tag{14}$$

This shows that the innovation term in equation (5) is the inverse of the Hessian matrix of the second order derivatives of the CM performance measure with respect to the vector θ multiplied by the gradient of the same function. Hence, the combination of equations (4) and (5) forms Newton's method recurrence formula in terms of the augmented parameter vector θ.

Now, observe that equation (6) is an output equation only. It is used to extract instantaneous values of the parameter vector W only and does not enter into the feedback loop for calculating the next iterate for θ. As a result, equation (6) does not affect the dynamic properties of the iterative process. The same is true for equations (7) and (8) when real data is used instead. Thus, the algorithm defined by equations (4), (5) and (6) or (7) and the one defined by equations (4), (5) and (8) are Newton's methods for searching the CM cost function in the space spanned by θ. This completes the proof of the theorem.

The theorem introduces a new algorithm for minimizing the CM performance measure. Unlike the original CMA algorithm where a formula for the sought after optimum solution is not known, this new algorithm seeks the optimum solution θ_{opt} which makes the gradient of equation (12) equal to zero. This solution is given by:

$$\theta_{opt} = R_{\varphi\varphi}^{-1} P_\varphi \tag{15}$$

It should be emphasized here that the theorem searches the function of equation (10) and not that of equation (2). These two functions are generally not equal except in the special case when the augmented vector θ can be decomposed into a kronecker product form. In fact, there is a gap between the minimum values of expressions (2) and (10). This gap goes to zero only when the estimated signal $\{y_t\}$ matches perfectly the unknown signal $\{s_t\}$. We argue here, however, that if the signal model is adequate, this gap is small enough. In this case, the vector W obtained by the theorem is close to that of the global minimum of the CM function. This can be seen by letting W_* be the parameter vector extracted form θ_{opt} using either equation (6), (7) or (8), and defining a vector θ_* as:

$$\theta_* = W_* \otimes \overline{W}_*$$

If θ_{opt} is a kronecker product form, then θ_* is equal to θ_{opt}. Thus, the values of expressions (2) and (11) are the same. In other words, the vector W_* operates at the absolute minimum of the CM performance measure $\mathcal{J}$. However, if θ_{opt} is not a kronecker product form, then the vector θ_* is different from, but close to the optimum vector θ_{opt}. In fact, θ_* is the closest vector to θ_{opt} that can be decomposed into a kronecker product form. Notice that while the vector θ_{opt} solves the system of equations in (12) when equated to zero, the extracted vector θ_* solves the system of equations:

$$R_{\varphi\varphi}\theta_* = P_*$$

The vector W_c at the global minimum of the CM function on the other hand verifies the system of equations:

$$R_{\varphi\varphi}\theta_c = P_c$$

where the vector θ_c is defined as:

$$\theta_c = W_c \otimes \overline{W}_c$$

Using perturbation theory [6], the norm of the difference between the two solutions θ_{opt} and θ_* is bounded as:

$$\frac{\| \theta_{opt} - \theta_* \|}{\| \theta_{opt} \|} \leq \varepsilon \mathcal{K}(R_{\xi\xi}) \frac{\| P_* - P_\varphi \|}{\| P_\varphi \|}$$

where ε is an arbitrarily small variable and $\mathcal{K}(R_{\xi\xi})$ is defined as $\mathcal{K}(R_{\xi\xi}) = \frac{\lambda_{max}(R_{\xi\xi})}{\lambda_{min}(R_{\xi\xi})}$, with $\lambda_{max}(R_{\xi\xi})$ and $\lambda_{min}(R_{\xi\xi})$ being the maximum and the minimum eigenvalues of the matrix $R_{\xi\xi}$, respectively.
Similarly, the norm of the difference between the two solutions θ_{opt} and θ_c is bounded as:

$$\frac{\| \theta_{opt} - \theta_c \|}{\| \theta_{opt} \|} \leq \varepsilon \mathcal{K}(R_{\xi\xi}) \frac{\| P_c - P_\varphi \|}{\| P_\varphi \|}$$

These differences are small when the signal model is adequate and the noise is not excessive.

3. IMPLEMENTATION ISSUES

The operations of the new algorithm for searching the CM function are straightforward. At each iteration of time k, the new algorithm computes a new value $\theta(k)$ of an augmented parameter vector using its previous value $\theta(k-1)$ and the received sequence statistics P_φ and $R_{\varphi\varphi}$. Then, it uses the value $\theta(k)$ to calculate a new value $W(k)$ of the filter's parameter vector. The process is repeated until the update term vanishes and the optimum solution is reached. At the start, a known initial value $\theta(-1)$ of the augmented parameter vector is required to proceed. Unlike the case of the original CMA algorithm, the null vector can be used as an initial condition if a better choice for $\theta(-1)$ is not available. This is true because the update term of the new algorithm is not equal to zero for a null vector as seen from equation (4).

The new algorithm is different from the method of Newton when used to search the standard mean square error function in three ways. First, the proposed algorithm performs the iterations on an augmented parameter vector θ instead of the actual parameter vector W directly. The number of components of the augmented vector θ is equal to the square of the number of components of the parameter vector W. This means that the new algorithm is much more computationally intensive than Newton's method applied to the standard mean square error minimization. Second, the new algorithm uses the second and the fourth order statistics P_φ and $R_{\varphi\varphi}$ of the received sequence instead of the usual second order statistics needed for Wiener filter. Again, this makes the new algorithm more computationally expensive than Newton's method when used to obtain Wiener solution. Finally, the proposed algorithm differs from Newton's method for searching the mean square cost function by using an output block $C(.)$ to compute the values of the parameter vector $W(k)$ from those of the augmented parameter vector $\theta(k)$. The output block $C(.)$ can be viewed as a measurement block in the same way as used in control systems and Kalman filtering. However, the output block $C(.)$ in our scheme is a nonlinear vector function while the measurement block in Kalman filtering is generally a simple constant coefficients matrix.

The new algorithm can be implemented in several ways. For example, observe that equation (5) provides a clear and explicit description of the parameter $\theta(k)$ in terms of the parameter $\theta(k-1)$. This is very useful for theoretical purposes. However, this equation is generally never used in practice. Instead, it is rewritten as:

$$R_{\varphi\varphi}(\theta(k) - \theta(k-1)) = -\mu\nabla(k-1) \tag{16}$$

Equation (16) is better for computation than equation (5) since it does not require the explicit calculation of the inverse of the matrix $R_{\varphi\varphi}$. Instead, the system of equation (16) is solved directly using efficient techniques such as LU decomposition or other methods.

This algorithm can also be implemented as a block optimization procedure. In this case, there is a preamble period of duration N needed for the iterations of equations (4) and (16) to converge to an acceptable value of the optimum augmented parameter vector θ_{opt} on the basis of the statistics P_φ and $R_{\varphi\varphi}$ of the received sequence $\{x_t\}$, with the time $t \leq N$. Once a reasonable estimate of θ_{opt} is reached, we proceed with equation (6), (7) or (8) to estimate W_{opt} and y_t on the basis of this solution and the sample vector X_t, with $t \geq N$. Note that no estimation of W_{opt} or of y_t is performed prior to time N. This block optimization is most effective when implemented in two separate chips. The first chip hosts equations (4) and (16) and runs at rate that is much faster than the communications link. The second chip performs the operations of equation (6), (7) or (8) and runs at the same rate as the communications link. The rate of the first chip is such that the augmented optimum solution θ_{opt} is reached before the next cycle of the communications link comes around.

Parallel or systolic arrays implementations of the algorithm of theorem 1 can also be devised to lessen the computational burden of this algorithm. These methods will, however, not be presented in this paper. Instead, we focus on the convergence properties of the new algorithm. It is hoped that a better dynamical behavior will offset the additional algorithmic complexity and computational demands introduced by the new algorithm.

Notice also that when the quantities P_φ and $R_{\varphi\varphi}$ are known a priori, one can proceed directly to determine the optimum weights and the optimum output using equations (4) through (8). However, these statistics may not be available in practice. In this case, they can be estimated using the $n \times 1$ vector X_t of the sample values as follows:

$$\widehat{P}_\varphi = \frac{1}{M} \sum_{t=0}^{M-1} \varphi_t \qquad (17)$$

$$\widehat{R}_{\varphi\varphi} = \frac{1}{M} \sum_{t=0}^{M-1} \varphi_t \varphi_t^* \qquad (18)$$

where the symbol M represents the data length used for the estimation of the statistics, and the augmented data vector φ_t is obtained from the sample vector X_t as defined earlier.

Finally, observe that the estimates of equations (17) and (18) are both unbiased and consistent. The calculations of these estimate statistics can also be added as a third stage to the front of the procedure described earlier. In this case, there is a first preamble period of duration M used to evaluate the statistics of expressions (17) and (18) on the basis of the observed samples of the received sequence $\{x_t\}$, with $t \leq M$. Once the estimates $\widehat{P}_\varphi$ and $\widehat{R}_{\varphi\varphi}$ are obtained, we proceed with the iterations of equations (4) and (16). As seen earlier, there is a second preamble period of duration N used to obtain an appropriate value the augmented optimum parameter vector θ_{opt} on the basis of the estimated statistics of the received sequence, with $M \leq t \leq N$. Then, estimates of W_{opt} and y_t on the basis of $\{x_t\}$, with $t \geq N$ are finally obtained. Again, this block optimization procedure is most effective when each of these three stages is implemented on a separate chip with an adequate rate.

4. CONCLUSIONS

The CMA algorithm has emerged as the method of choice in blind adaptive equalization in recent years. This paper has introduced a new iterative algorithm for minimizing the CM performance function using Newton's method of search. Unlike the original CMA algorithm, the new iterative procedure has the advantage of converging to a desired solution only. Expressed in terms of the statistics of the sample sequence only, the new algorithm can also be used with other applications besides equalization.

5. REFERENCES

[1] D. N. Godard, "Self-Recovering Equalization and Carrier Tacking in Two-Dimensional Data Communication Systems," *IEEE Transaction on Communication,* vol. COM-28, 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 Acoustic, Speech and Signal Processing,* Vol. ASSP-31, No. 2, pp. 459-472, April 1983.

[3] C. R. Johnson Jr., P. Schniter, T. J. Endres, J. D. Behm, D. R. Brown, and R. A. Casas "Blind Equalization Using the Constant Modulus Criterion: A Review," *Proceedings of the IEEE,* Special Issue on Blind System Identification and Estimation, pp. 1927-1950, Oct. 1998.

[4] Z. Ding, R.A. Kennedy, B.D.O. Anderson, and C.R. Johnson, Jr. "On the (non)Existence of Undesirable equilibria of Godard Blind Equalizer," *IEEE Transactions on Signal Processing,* Vol. 40, pp. 2425-2432, Oct. 1992.

[5] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics,* New York, John Wiley and Sons, 1988, Reprinted April 1990.

[6] G. H. Golub and C. F. Van Loan, "Matrix Computations," 3^{rd} edition, Baltimore, John Hopkins University Press, 1996.