On a Class of Computationally Efficient, Rapidly Converging, Generalized NLMS Algorithms

Dennis R. Morgan, *Senior Member, IEEE,* and Steven G. Kratzer, *Member, IEEE*

***Abstract*— Over the last decade, a certain computationally efficient, rapidly converging adaptive filtering algorithm has been independently discovered many times. The algorithm can be viewed as a generalization of the normalized LMS (NLMS) algorithm that updates on the basis of multiple input signal vectors. This correspondence compares and discusses the different approaches to and embellishments of the basic algorithm, and contrasts the various interpretations from different perspectives.**

I. Introduction

We first establish the necessary background in terms of the conventional LMS and normalized LMS (NLMS) algorithms leading to the basic form of the generalized NLMS (GNLMS) algorithm. We then comment on the history of its many independent discoveries and contrast the varied approaches used in the derivation and embellishments of the basic form, all of which are variations on the same theme. The contrasting interpretations illuminate the problem from different perspectives, providing a rich composite viewpoint.

Consider an adaptive finite impulse response (FIR) filter that produces an output

$$y(n) = \mathbf{w}^T(n)\mathbf{x}(n) \tag{1}$$

where $\mathbf{w}(n) \equiv [w_0(n) \quad w_1(n) \cdots w_{L-1}(n)]^T$ is the vector of adaptive filter coefficients, the superscript T denotes transpose, and $\mathbf{x}(n) \equiv [x(n) \quad x(n-1) \cdots x(n-L+1)]^T$ is a vector of the most recent L samples of the input signal $x(n)$. Let $d(n)$ denote the "desired" signal and form the error signal as

$$e(n) = d(n) - y(n). \tag{2}$$

The standard LMS (stochastic gradient) algorithm updates the adaptive weight vector by the gradient of the instantaneous squared error [1] as

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\mathbf{x}(n)e(n) \tag{3}$$

where μ is the step size. It is well known that when the input signal is highly correlated, i.e., when the covariance matrix of $\mathbf{x}(n)$ has a large range of eigenvalues, the LMS algorithm can suffer from slow convergence.

Manuscript received January 30, 1996. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. J. J. Shynk.

D. R. Morgan is with the Acoustics and Audio Communication Research Deptartment, Bell Laboratories, Lucent Technologies, Murray Hill, NJ 07974-0636 USA (e-mail: drrm@research.att.com).

S. G. Kratzer is with the Center for Computing Sciences, Bowie, MD 20715 USA.

Publisher Item Identifier S 1070-9908(96)06509-1.

A normalized form of the LMS algorithm is often employed, which is written

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \alpha\mathbf{x}(n)[\mathbf{x}^T(n)\mathbf{x}(n)]^{-1}e(n) \tag{4}$$

where the normalized step size α is usually somewhat less than or equal to one. The normalized LMS (NLMS) algorithm can be understood from at least three viewpoints. The simplest interpretation [2] is that (4) is merely a way of making the convergence rate independent of signal power by letting $\mu = \alpha/[L\hat{P}_x(n)]$ in (3), where $\hat{P}_x(n) \equiv \mathbf{x}^T(n)\mathbf{x}(n)/L$ is a running-average estimate of the instantaneous signal power $E[x^2(n)]$. From this viewpoint, no significant difference would be expected between the convergence performance of the LMS and NLMS algorithms for stationary signals provided that the step sizes are appropriately selected; however, for nonstationary signals like speech, the NLMS can provide significantly faster convergence for the same level of steady-state misadjustment [3]. Another interpretation of the NLMS algorithm is as the limiting result of reiterating the LMS algorithm many times at each time step using the same data [4], i.e., the updated weight $\mathbf{w}(n+1)$ from (3) is repeatedly substituted for $\mathbf{w}(n)$ in (1), and (2) and (3) are recalculated. Finally, for $\alpha = 1$, the NLMS algorithm can be viewed as the solution to a least-squares problem that determines $\mathbf{w}(n+1)$ as the vector closest to $\mathbf{w}(n)$ such that $\mathbf{w}^T(n+1)\mathbf{x}(n) = d(n)$ [5, p. 416], and can be interpreted [6] as a projection of $\mathbf{w}(n)$ that solves an underdetermined least-squares problem. It is this last viewpoint that first inspired the basic form of the GNLMS algorithm, which we discuss in the next section.

An anonymous reviewer also provided a constrained version of the least-squares approach, which first chooses $\Delta\mathbf{w}$ to minimize $||e(n) - \mathbf{x}^T(n)\Delta\mathbf{w}||^2 + \beta||\Delta\mathbf{w}||^2$, where β is a Lagrange multiplier, and then forms the relaxed update

$$\begin{aligned}\mathbf{w}(n+1) &= \mathbf{w}(n) + \alpha\Delta\mathbf{w} \\ &= \mathbf{w}(n) + \alpha\mathbf{x}(n)[\mathbf{x}^T(n)\mathbf{x}(n) + \beta]^{-1}e(n).\end{aligned} \tag{5}$$

This form is, in fact, often used heuristically in practice to limit the maximum step size of the NLMS algorithm so that small values of $\mathbf{x}^T\mathbf{x}$ do not cause the algorithm to go unstable.

II. Generalized NLMS Algorithms

In [7], Ozeki and Umeda develop the basic form of the GNLMS algorithm from the geometric viewpoint of affine subspace projections that, in the noiseless case, require the updated adaptive weight to be as close as possible to the previous weight vector while producing zero error for each

of a block of N input vectors (hard constraint). The relaxed form of this process is expressed as

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \alpha \mathbf{X}(n)[\mathbf{X}^T(n)\mathbf{X}(n)]^{-1}\mathbf{e}(n) \quad (6)$$

where

$$\mathbf{X}(n) \equiv [\mathbf{x}(n) \quad \mathbf{x}(n-1) \cdots \mathbf{x}(n-N+1)] \quad (7)$$

is the matrix formed by a block of N input signal vectors, with associated error vector

$$\mathbf{e}(n) \equiv [e(n) \quad e(n-1) \cdots e(n-N+1)]^T. \quad (8)$$

In this and other formulations in this paper, the intention is that the block length N is smaller than the adaptive filter order L. This was clearly the intention in [7]. However, their solution is written in terms of the Moore–Penrose pseudoinverse of $\mathbf{X}^T$, which unfortunately was not explicitly defined in [7]. For the underdetermined case $N < L$, this pseudoinverse is written as $(\mathbf{X}^T)^\dagger \equiv \mathbf{X}(n)[\mathbf{X}^T(n)\mathbf{X}(n)]^{-1}$, and this is the form that appears above in (6), which corresponds to a minimum-norm solution. However, the pseudoinverse is also defined for the overdetermined case $N \geq L$ as $(\mathbf{X}^T)^\dagger \equiv [\mathbf{X}(n)\mathbf{X}^T(n)]^{-1}\mathbf{X}(n)$, and so their formulation could also be interpreted in that case as a least-squares solution over a block of N errors.

Ozeki and Umeda call (6) the *affine projection algorithm* (APA) after the method of its derivation. Their work establishes the fundamental performance advantage of the algorithm, by which rapid convergence is obtained for correlated signals that can be modeled by an autoregressive (AR) process of order less than or equal to N. Note that for $N = 1$, (6) reduces to the NLMS algorithm (4), and so can be considered as a generalization of the NLMS algorithm.

Although a performance advantage was established in [7], the above form of the algorithm requires a great deal of computation. Also, there is some concern for numerical conditioning due to the possible near-singularity of $\mathbf{X}^T(n)\mathbf{X}(n)$. A variation of the basic GNLMS algorithm called the *partial rank algorithm* (PRA) was independently discovered by Kratzer and Morgan [8], which addresses these problems. (Although the PRA was developed within the context of spatial processing for adaptive arrays, the algorithm of course also applies to temporal processing, where the input signal vector consists of contiguous time samples.) First of all, it was recognized that little performance is sacrificed if $\mathbf{w}(n)$ is updated only every N samples instead of every sample, thereby realizing significant computational savings. Also, a small regularization constant can be added to the diagonal of $\mathbf{X}^T(n)\mathbf{X}(n)$ for numerical conditioning. The PRA was developed by starting with a stochastic version of the LMS/Newton method [1] with rank-deficient update, which is written

$$\mathbf{w}(n+1) = \mathbf{w}(n-N+1) + \alpha[\mathbf{X}(n)\mathbf{X}^T(n) + \beta\mathbf{I}]^{-1}\mathbf{X}(n)\mathbf{e}(n) \quad (9)$$

where it is understood that $\mathbf{w}(m) = \mathbf{w}(n-N+1)$, $m = n, n-1, \cdots, n-N+1$ remains unchanged in calculating (1), (2), and (8). Using Woodbury's identity (the matrix inversion lemma [5, p. 480]) yields the mathematically equivalent form

$$\mathbf{w}(n+1) = \mathbf{w}(n-N+1) + \alpha\mathbf{X}(n)[\mathbf{X}^T(n)\mathbf{X}(n) + \beta\mathbf{I}]^{-1}\mathbf{e}(n) \quad (10)$$

and this expression constitutes the PRA. In the application of (10), the values of α and β can be optimized so that fastest convergence is achieved for a given steady-state misadjustment [8]. Of course, for $\beta = 0$, the regularized algorithm reduces to a block update version of (6), meaning that $\mathbf{w}$ is updated after every N samples instead of after every sample. Kratzer and Morgan [8] also note that the PRA algorithm, being of order NL complexity (per sample time), provides a link between computationally intensive (full-rank) methods like sample matrix inversion (SMI) [9] (order L^2 complexity) and the simple LMS algorithm (order L complexity), providing for graceful performance degradation as the underlying input signal model (rank of covariance matrix) increases in complexity (see [8] for details).

Interestingly, (10) can be alternatively viewed as the outcome of three other very different approaches [8]: i) spatial Wiener/inverse filtering; ii) minimizing the error vector over a block of input vectors (as in [7]) while simultaneously requiring that the weight vector "not change too much" (soft constraint); and iii) minimizing the error by simultaneously varying both the desired signal vector and the weight vector. Whichever of these three derivations is used, update (9) or (10) has the effect of choosing $\Delta\mathbf{w}$ to minimize $||\mathbf{e}(n) - \mathbf{X}^T(n)\Delta\mathbf{w}||^2 + \beta||\Delta\mathbf{w}||^2$ and forming the relaxed update $\mathbf{w}(n+1) = \mathbf{w}(n-N+1) + \alpha\Delta\mathbf{w}$. This interpretation is a generalization of the modified NLMS algorithm (5), as previously discussed. An anonymous reviewer also provided another viewpoint, whereby the same result is achieved by minimizing $\alpha||\mathbf{e}(n) - \mathbf{X}^T(n)\Delta\mathbf{w}||^2 + (1-\alpha)||\mathbf{X}^T(n)\Delta\mathbf{w}||^2 + \beta||\Delta\mathbf{w}||^2$, which can be interpreted as linear interpolation of two quadratic forms as α goes between zero and one. We further note that approach ii) above is one of the two cases of Slock's "unified criteria" [10] (the second one he mentions has $L > 1$ samples and our $\beta = 0$); however, his explicit sliding window uses recursive updating of $(\mathbf{X}\mathbf{X}^T)^{-1}(\beta = 0)$ rather than utilizing a regularization term $(\beta \neq 0)$.

In [11], Wang and Wang independently derive the unrelaxed $(\alpha = 1)$ and unregularized $(\beta = 0)$ form of (10) as the solution to a minimum-norm least-squares problem, emphasizing the nature of the problem as a minimization of the *a posteriori* estimation error. They interpret the algorithm on the basis of projecting the weight misalignment onto a subspace of dimension $L-N$ determined by N input signal vectors, being very similar to the approach of [7]. They also relate the form of their algorithm to an "optimal" block LMS algorithm that uses optimally determined step sizes for each component in the block. Accordingly, they call their algorithm the *generalized optimal block algorithm* or GOBA. They also point out the equivalency of their method to block recursive least squares for $N \geq L$,, which is analogous to the equivalency of (9) and (10) for the more general case $(\alpha \neq 1, \beta \neq 0)$.

The most recent discovery of the algorithm appears in [12] by Montazeri and Duhamel. Their approach is to assume

that the weight update is formed as a linear combination of instantaneous gradients associated with N input vectors. They then calculate the minimum-norm weight update that nulls the N *a posteriori* errors, which upon relaxation arrives at the unregularized ($\beta = 0$) form of (10). Thus, the formulation in [12] is a relaxed version of [11] and a special case of the general formulation (10). They also discuss convergence of the mean weight and mean-square error, but do not provide general quantitative results analogous to those of [9] (showing, for example, that SMI with $N = 2L$ samples gives approximately 3 dB of misadjustment.) They also employ fast block methods to reduce the computational complexity.

Finally, in [13] and [14], a different approach is taken to reduce the computational complexity of (6). Instead of block updating every L samples, they implement a regularized version of (6) that updates on a sample-by-sample basis, i.e.,

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \alpha\mathbf{X}(n)[\mathbf{X}^T(n)\mathbf{X}(n) + \beta\mathbf{I}]^{-1}\mathbf{e}(n) \quad (11)$$

but use computationally efficient techniques similar to those employed in fast versions of the recursive least-squares (RLS) algorithm that exploit redundancies in the data structure. In this way, they achieve a computational complexity on the order of $2L$, which for $N > 2$ is actually less than that of the block update (10). Moreover, the computational load of their method is more uniformly distributed in time and, therefore, makes for more efficient implementation with digital signal processing architectures. They call their algorithm the "fast affine projection" (FAP) algorithm and "fast projection" algorithm, respectively, after the affine projection algorithm of [7], even though their formulation is a little more general due to the regularization term β in (11).

III. Conclusion

We hope that the forgoing discussion will provide a general appreciation of the GNLMS algorithm from its many different perspectives. The discussion also highlights the opportunity for further study of the algorithm performance characteristics.

Acknowledgment

The authors wish to express their thanks to S. L. Gay for useful discussions in the course of developing this work.

References

[1] B. Widrow and S. D. Stearns, *Adaptive Signal Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1985.
[2] J. R. Treichler, C. R. Johnson, Jr., and M. G. Larimore, *Theory and Design of Adaptive Filters.* New York: Wiley, 1987, p. 83.
[3] M. Rupp, "The behavior of LMS and NLMS algorithms in the presence of spherically invariant processes," *IEEE Trans. Signal Processing,* vol. 41, pp. 1149–1160, Mar. 1993.
[4] R. Nitzberg, "Application of the normalized LMS algorithm to MSLC," *IEEE Trans. Aerosp. Electron. Syst.,* vol. AES-21, pp. 79–91, Jan. 1985.
[5] S. Haykin, *Adaptive Filter Theory.* Englewood Cliffs, NJ: Prentice-Hall, 1991.
[6] G. C. Goodwin and K. S. Sin, *Adaptive Filtering Prediction and Control.* Englewood Cliffs, NJ: Prentice-Hall, 1984, ch. 3.
[7] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *Electron. Commun. Japan,* vol. 67-A, pp. 19–27, 1984.
[8] S. G. Kratzer and D. R. Morgan, "The partial-rank algorithm for adaptive beamforming," in *Proc. SPIE Int. Soc. Optic. Eng.,* 1985, vol. 564, pp. 9–14.
[9] I. S. Reed, J. D. Mallett, and L. E. Brennan, "Rapid convergence rate in adaptive arrays," *IEEE Trans. Aerosp. Electron. Syst.,* vol. AES-10, pp. 853–863, Nov. 1974.
[10] D. T. M. Slock, "Underdetermined growing and sliding window covariance fast transversal filter RLS algorithm," in *Proc. EUSIPCO-92,* 1992, vol. II, pp. 1169–1172.
[11] T. Wang and C.-L. Wang, "On the optimum design of the block adaptive FIR digital filter," *IEEE Trans. Signal Processing,* vol. 41, pp. 2131–2140, June 1993.
[12] M. Montazeri and P. Duhamel, "A set of algorithms linking NLMS and block RLS algorithms," *IEEE Trans. Signal Processing,* vol. 43, pp. 444–453, Feb. 1995.
[13] S. L. Gay and S. Tavathia, "The fast affine projection algorithm," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* 1995, pp. 3023–3026.
[14] M. Tanaka, Y. Kaneda, S. Makino, and J. Kojima, "Fast projection algorithm and its step size control," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* 1995, pp. 945–948.