THE LMS, PNLMS, AND EXPONENTIATED GRADIENT ALGORITHMS

Jacob Benesty[1] *and Yiteng (Arden) Huang*[2]

[1]Université du Québec, INRS-EMT
800 de la Gauchetière Ouest, Suite 6900
Montréal, Québec, H5A 1K6, Canada
[2]Bell Labs, Lucent Technologies
600 Mountain Avenue
Murray Hill, NJ, 07974, USA
e-mail: benesty@inrs-emt.uquebec.ca, yitenghuang@bell-labs.com
web: http://www.inrs-telecom.uquebec.ca/users/benesty/

ABSTRACT

Sparse impulse responses are encountered in many applications (network and acoustic echo cancellation, feedback cancellation in hearing aids, etc). Recently, a class of exponentiated gradient (EG) algorithms has been proposed. One of the algorithms belonging to this class, the so-called EG± algorithm, converges and tracks much better than the classical stochastic gradient, or LMS, algorithm for sparse impulse responses. In this paper, we show how to derive the different algorithms. We analyze the EG± algorithm and explain when to expect it to behave like the LMS algorithm. It is also shown that the proportionate normalized LMS (PNLMS) algorithm proposed by Duttweiler in the context of network echo cancellation is an approximation of the EG±.

1. INTRODUCTION

One of the most popular adaptive algorithms available in the literature is the stochastic gradient algorithm also called least-mean-square (LMS) [1], [2]. The main drawback of this algorithm is that it converges very slowly in general with correlated input signals.

Recently, another variant of the LMS algorithm, called the exponentiated gradient algorithm with positive and negative weights (EG± algorithm), was proposed by Kivinen and Warmuth [3]. This new algorithm converges much faster than the LMS algorithm when the impulse response that we need to identify is sparse, which is often the case in network echo cancellation involving a hybrid transformer in conjunction with variable network delay, or in the context of hands-free communications where there is a strong coupling between the loudspeaker and the microphone [4]. The EG± algorithm has the nice feature that its update rule takes advantage of the sparseness of the impulse response to speed up its initial convergence and to improve its tracking abilities compared to LMS. More recently, a technique known as the proportionate normalized LMS (PNLMS) algorithm [5] has been introduced which has similar advantages for sparse impulse responses. In [6], a general expression of the mean squared error (MSE) is derived for the EG± algorithm showing that for sparse impulse responses, the EG± algorithm, like PNLMS, converges more quickly than the LMS for a given asymptotic MSE.

In this paper, we show how to derive several important algorithms. We explain some interesting links between the LMS and EG± algorithms, when to expect them to behave in the same way and that the choice of some parameters of the EG± is critical. We also show that the PNLMS algorithm is an approximation of the EG± algorithm.

2. DERIVATION OF THE DIFFERENT ALGORITHMS

In this section, we show how to derive different variants of the LMS algorithm. Depending on how we define the distance between the old and new weight vectors, we obtain different update rules.

We define the *a priori* error signal $e(n+1)$ at time $n+1$ as:

$$e(n+1) = y(n+1) - \hat{y}(n+1), \tag{1}$$

where

$$y(n+1) = \mathbf{h}_{\mathrm{t}}^T \mathbf{x}(n+1) \tag{2}$$

is the system output,

$$\mathbf{h}_{\mathrm{t}} = [\ h_{\mathrm{t},0} \quad h_{\mathrm{t},1} \quad \cdots \quad h_{\mathrm{t},L-1}\ ]^T$$

is the true (subscript t) impulse response of the system, superscript ^{T} denotes transpose of a vector or a matrix,

$$\mathbf{x}(n+1) = [\ x(n+1) \quad x(n) \quad \cdots \quad x(n-L+2)\ ]^T$$

is a vector containing the last L samples of the input signal x,

$$\hat{y}(n+1) = \mathbf{h}^T(n)\mathbf{x}(n+1), \tag{3}$$

is the model filter output, and

$$\mathbf{h}(n) = [\ h_0(n) \quad h_1(n) \quad \cdots \quad h_{L-1}(n)\ ]^T$$

is the model filter.

One easy way to find adaptive algorithms that adjust the new weight vector $\mathbf{h}(n+1)$ from the old one $\mathbf{h}(n)$ is to minimize the following function [3]:

$$J[\mathbf{h}(n+1)] = d[\mathbf{h}(n+1), \mathbf{h}(n)] + \eta\varepsilon^2(n+1), \tag{4}$$

where $d[\mathbf{h}(n+1), \mathbf{h}(n)]$ is some measure of distance from the old to the new weight vector,

$$\varepsilon(n+1) = y(n+1) - \mathbf{h}^T(n+1)\mathbf{x}(n+1) \tag{5}$$

is the *a posteriori* error signal, and η is a positive constant. The magnitude of η represents the importance of correctiveness compared to the importance of conservativeness [3]. If η is very small, minimizing $J[\mathbf{h}(n+1)]$ is close to minimizing $d[\mathbf{h}(n+1),\mathbf{h}(n)]$, so that the algorithm makes very small updates. On the other hand, if η is very large, the minimization of $J[\mathbf{h}(n+1)]$ is almost equivalent to minimizing $d[\mathbf{h}(n+1),\mathbf{h}(n)]$ subject to the constraint $\varepsilon(n+1)=0$.

To minimize $J[\mathbf{h}(n+1)]$, we need to set its L partial derivatives $\partial J[\mathbf{h}(n+1)]/\partial h_l(n+1)$ to zero. Hence, the different weight coefficients $h_l(n+1)$, $l=0,1,...,L-1$, will be found by solving the equations:

$$\frac{\partial d[\mathbf{h}(n+1),\mathbf{h}(n)]}{\partial h_l(n+1)} - 2\eta x(n+1-l)\varepsilon(n+1) = 0. \qquad (6)$$

Solving (6) is in general very difficult. However, if the new weight vector $\mathbf{h}(n+1)$ is close to the old weight vector $\mathbf{h}(n)$, replacing the *a posteriori* error signal $\varepsilon(n+1)$ in (6) with the *a priori* error signal $e(n+1)$ is a reasonable approximation and the equation

$$\frac{\partial d[\mathbf{h}(n+1),\mathbf{h}(n)]}{\partial h_l(n+1)} - 2\eta x(n+1-l)e(n+1) = 0 \qquad (7)$$

is much easier to solve for all distance measures d.

The LMS algorithm is easily obtained from (7) by using the squared Euclidean distance

$$d_{\mathrm{E}}[\mathbf{h}(n+1),\mathbf{h}(n)] = \|\mathbf{h}(n+1)-\mathbf{h}(n)\|_2^2. \qquad (8)$$

The exponentiated gradient (EG) algorithm with positive weights results from using for d the *relative entropy*, also known as *Kullback-Leibler divergence*,

$$d_{\mathrm{re}}[\mathbf{h}(n+1),\mathbf{h}(n)] = \sum_{l=0}^{L-1} h_l(n+1)\ln\frac{h_l(n+1)}{h_l(n)}, \qquad (9)$$

with the constraint $\sum_l h_l(n+1)=1$, so that (7) becomes:

$$\frac{\partial d_{\mathrm{re}}[\mathbf{h}(n+1),\mathbf{h}(n)]}{\partial h_l(n+1)} - 2\eta x(n+1-l)e(n+1) + \gamma = 0, \quad (10)$$

where γ is the Lagrange multiplier. Actually, the appropriate constraint should be $\sum_l h_l(n+1) = \sum_l h_{\mathrm{t},l}$ but $\sum_l h_{\mathrm{t},l}$ is not known in practice, so we take the arbitrary value 1 instead. This will have an effect on the adaptation step of the resulting adaptive algorithm.

The algorithm derived from (10) is valid only for positive weights. To deal with both positive and negative coefficients, we can always find two vectors $\mathbf{h}^+(n+1)$ and $\mathbf{h}^-(n+1)$ with positive coefficients, in such a way that the vector

$$\mathbf{h}(n+1) = \mathbf{h}^+(n+1) - \mathbf{h}^-(n+1) \qquad (11)$$

can have positive and negative components. In this case, the *a posteriori* error signal can be written as:

$$\varepsilon(n+1) = y(n+1) - [\mathbf{h}^+(n+1) - \mathbf{h}^-(n+1)]^T\mathbf{x}(n+1) \qquad (12)$$

and the function (4) will change to:

$$\begin{aligned} J[\mathbf{h}^+(n+1),\mathbf{h}^-(n+1)] &= \qquad (13)\\ d[\mathbf{h}^+(n+1),\mathbf{h}^+(n)] &+ d[\mathbf{h}^-(n+1),\mathbf{h}^-(n)]\\ +\frac{\eta}{u}\varepsilon^2(n+1), \end{aligned}$$

where u is a positive scaling constant. Using the same approximation as before and choosing the Kullback-Leibler divergence plus the constraint $\sum_l [h_l^+(n+1) + h_l^-(n+1)] = u$, the solutions of the equations

$$\frac{\partial d_{\mathrm{re}}[\mathbf{h}^+(n+1),\mathbf{h}^+(n)]}{\partial h_l^+(n+1)} - 2\frac{\eta}{u}x(n+1-l)e(n+1) + \gamma = 0,$$

$$\frac{\partial d_{\mathrm{re}}[\mathbf{h}^-(n+1),\mathbf{h}^-(n)]}{\partial h_l^-(n+1)} + 2\frac{\eta}{u}x(n+1-l)e(n+1) + \gamma = 0,$$

give the so-called EG± algorithm, where

$$e(n+1) = y(n+1) - [\mathbf{h}^+(n) - \mathbf{h}^-(n)]^T\mathbf{x}(n+1), \qquad (14)$$

and will be further detailed in the next section.

When the parameter space is a curved manifold (non Euclidean), there are no orthonormal linear coordinates and the squared length of a small incremental vector $\mathbf{h}(n+1)-\mathbf{h}(n)$ connecting $\mathbf{h}(n)$ and $\mathbf{h}(n+1)$ is given by the quadratic form:

$$\begin{aligned} &d_{\mathrm{R}}[\mathbf{h}(n+1),\mathbf{h}(n)] = \\ &[\mathbf{h}(n+1)-\mathbf{h}(n)]^T\mathbf{G}[\mathbf{h}(n)][\mathbf{h}(n+1)-\mathbf{h}(n)]. \quad (15) \end{aligned}$$

Such a space is a Riemannian space. The $L \times L$ positive-definite matrix $\mathbf{G}[\mathbf{h}(n)]$ is called the *Riemannian metric tensor* and it depends in general on $\mathbf{h}(n)$. The Riemannian metric tensor characterizes the intrinsic curvature of a particular manifold in L-dimensional space. In the Euclidean orthonormal case, $\mathbf{G}[\mathbf{h}(n)] = \mathbf{I}$ (the identity matrix) and (15) is the same as (8). Using (15) in (7), we obtain the natural gradient descent algorithm proposed by Amari [7]:

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \eta\mathbf{G}^{-1}[\mathbf{h}(n)]\mathbf{x}(n+1)e(n+1). \qquad (16)$$

3. LINK BETWEEN THE LMS AND EG± ALGORITHMS

Let us define the LMS algorithm [1]:

$$\begin{aligned} e(n+1) &= y(n+1) - \mathbf{h}^T(n)\mathbf{x}(n+1), \qquad (17)\\ \mathbf{h}(n+1) &= \mathbf{h}(n) + \mu\mathbf{x}(n+1)e(n+1). \qquad (18) \end{aligned}$$

If we initialize $h_l(0)=0$, $l=0,1,...,L-1$, we can easily see that:

$$\mathbf{h}(n+1) = \mu\sum_{i=0}^{n}\mathbf{x}(i+1)e(i+1). \qquad (19)$$

The EG± algorithm is:

$$e(n+1) = y(n+1) - [\mathbf{h}^+(n) - \mathbf{h}^-(n)]^T\mathbf{x}(n+1), \qquad (20)$$

$$h_l^+(n+1) = u\frac{h_l^+(n)r_l^+(n+1)}{\sum_{j=0}^{L-1}[h_j^+(n)r_j^+(n+1) + h_j^-(n)r_j^-(n+1)]}, \qquad (21)$$

$$h_l^-(n+1) = u\frac{h_l^-(n)r_l^-(n+1)}{\sum_{j=0}^{L-1}[h_j^+(n)r_j^+(n+1) + h_j^-(n)r_j^-(n+1)]}, \qquad (22)$$

where

$$\begin{aligned} r_l^+(n+1) &= \exp\left[\frac{\mu'}{u}x(n+1-l)e(n+1)\right], && (23) \\ r_l^-(n+1) &= \exp\left[-\frac{\mu'}{u}x(n+1-l)e(n+1)\right] \\ &= \frac{1}{r_l^+(n+1)}, && (24) \end{aligned}$$

and u is a constant chosen such that $u \geq \|\mathbf{h}_\mathrm{t}\|_1$. We can check that we always have $\|\mathbf{h}^+(n+1)\|_1 + \|\mathbf{h}^-(n+1)\|_1 = u$.

By exponentiating the update, the EG± algorithm has the effect of assigning larger relative updates to larger weights, thereby deemphasizing the effect of smaller weights. This is qualitatively similar to the PNLMS algorithm, to be described in more detail in the next section, which makes the update *proportional* to the size of the weight. This type of behavior is desirable for sparse impulse responses where small weights do not contribute significantly to the *mean* solution but introduce an undesirable noise-like *variance*.

Starting adaptation of the EG± algorithm with $h_l^+(0) = h_l^-(0) = c > 0,\ l = 0,1,...,L-1$, we can show that (21) and (22) are equivalent to:

$$\begin{aligned} h_l^+(n+1) &= u\frac{s_l^+(n+1)}{\sum_{j=0}^{L-1}[s_j^+(n+1)+s_j^-(n+1)]}, && (25) \\ h_l^-(n+1) &= u\frac{s_l^-(n+1)}{\sum_{j=0}^{L-1}[s_j^+(n+1)+s_j^-(n+1)]}, && (26) \end{aligned}$$

where

$$\begin{aligned} s_l^+(n+1) &= \exp\left[\frac{\mu'}{u}\sum_{i=0}^{n}x(i+1-l)e(i+1)\right], && (27) \\ s_l^-(n+1) &= \exp\left[-\frac{\mu'}{u}\sum_{i=0}^{n}x(i+1-l)e(i+1)\right] \\ &= \frac{1}{s_l^+(n+1)}. && (28) \end{aligned}$$

Clearly, the convergence of the algorithm does not depend of the initialization parameter c. Now

$$\begin{aligned} h_l(n+1) &= h_l^+(n+1) - h_l^-(n+1) && (29) \\ &= u\frac{s_l^+(n+1)-s_l^-(n+1)}{\sum_{j=0}^{L-1}[s_j^+(n+1)+s_j^-(n+1)]} \\ &= u\frac{\sinh\left[\frac{\mu'}{u}\sum_{i=0}^{n}x(i+1-l)e(i+1)\right]}{\sum_{j=0}^{L-1}\cosh\left[\frac{\mu'}{u}\sum_{i=0}^{n}x(i+1-j)e(i+1)\right]}. \end{aligned}$$

Note that the sinh function has the effect of exponentiating the update, as previously commented.

For u large enough and using the approximations $\sinh(a) \approx a$ and $\cosh(a) \approx 1$ when $|a| \ll 1$, (29) becomes:

$$h_l(n+1) = \frac{\mu'}{L}\sum_{i=0}^{n}x(i+1-l)e(i+1). \qquad (30)$$

We understand that, by taking $\mu' = L\mu$ and for u large enough, the LMS and EG± algorithms have the same performance. Obviously, the choice of u is critical in practice: if we take it too large, there is not a real advantage using the EG± algorithm.

Table 1 The improved proportionate NLMS algorithm.

Initialization:

$$h_l(0) = 0,\ l = 0,1,...,L-1$$

Parameters:

$$0 < \alpha \leq 1,\ \delta_{\mathrm{IPNLMS}} > 0,\ -1 \leq \kappa \leq 1$$

$\varepsilon > 0$ (small number to avoid division by zero)

Error:

$$e(n+1) = y(n+1) - \mathbf{h}^T(n)\mathbf{x}(n+1)$$

Update:

$$g_l(n) = \frac{1-\kappa}{2L} + (1+\kappa)\frac{|h_l(n)|}{2\|\mathbf{h}(n)\|_1+\varepsilon}$$

$$l = 0,1,...,L-1$$

$$\mu(n+1) = \frac{\alpha}{\sum_{j=0}^{L-1}x^2(n+1-j)g_j(n)+\delta_{\mathrm{IPNLMS}}}$$

$$h_l(n+1) = h_l(n) + \mu(n+1)g_l(n)x(n+1-l)e(n+1)$$

$$l = 0,1,...,L-1$$

4. LINK BETWEEN THE PNLMS AND EG± ALGORITHMS

Recently, the proportionate normalized least-mean-square (PNLMS) algorithm was developed for use in network echo cancelers [5]. In comparison to the NLMS algorithm, PNLMS has very fast initial convergence and tracking when the echo path is sparse. The idea behind PNLMS is to update each coefficient of the filter independently of the others by adjusting the adaptation step size in proportion to the magnitude of the estimated filter coefficient. More recently, an improved PNLMS (IPNLMS) [8] was proposed that performs better than NLMS and PNLMS, whatever the nature of the impulse response is. Table 1 summarizes the IPNLMS algorithm. In general, $g(l)$ in the table provides the "proportionate" scaling of the update. The parameter κ controls the amount of proportionality in the update. For $\kappa = -1$, it can easily be checked that the IPNLMS and NLMS algorithms are identical. For κ close to 1, the IPNLMS behaves like the PNLMS algorithm [5]. In practice, a good choice for κ is 0 or -0.5.

How are the IPNLMS and EG± algorithms specifically related? In the rest of this section, we show that the IPNLMS is in fact an approximation of the EG±.

For $|a| \ll 1$, we have: $\exp(a) \approx 1 + a$. For μ' small enough, the numerator and denominator of the EG± update equations can be approximated as follows:

$$r_l^+(n+1) \approx 1 + \frac{\mu'}{u}x(n+1-l)e(n+1), \qquad (31)$$

$$r_l^-(n+1) \approx 1 - \frac{\mu'}{u}x(n+1-l)e(n+1), \qquad (32)$$

$$\begin{aligned} &\sum_{j=0}^{L-1}[h_j^+(n)r_j^+(n+1) + h_j^-(n)r_j^-(n+1)] && (33) \\ &\approx u + \frac{\mu'}{u}\hat{y}(n+1)e(n+1) \approx u. \end{aligned}$$

With these approximations, (21) and (22) become:

$$h_l^+(n+1) = h_l^+(n)[1+\frac{\mu'}{u}x(n+1-l)e(n+1)], \quad (34)$$

$$h_l^-(n+1) = h_l^-(n)[1-\frac{\mu'}{u}x(n+1-l)e(n+1)], \quad (35)$$

so that:

$$\begin{aligned} h_l(n+1) &= h_l^+(n+1) - h_l^-(n+1) \quad (36) \\ &= h_l(n) + \mu' \frac{h_l^+(n)+h_l^-(n)}{\|\mathbf{h}^+(n)\|_1 + \|\mathbf{h}^-(n)\|_1} x(n+1-l)e(n+1). \end{aligned}$$

If the true impulse response $\mathbf{h}_\mathrm{t}$ is sparse, it can be shown that if we choose $u = \|\mathbf{h}_\mathrm{t}\|_1$, the (positive) vector $\mathbf{h}^+(n) + \mathbf{h}^-(n)$ is also sparse after convergence. This means that the elements $\frac{h_l^+(n)+h_l^-(n)}{\left\|\mathbf{h}^+(n)\right\|_1+\left\|\mathbf{h}^-(n)\right\|_1}$ in (36) play exactly the same role as the elements $g_l(n)$ in the IPNLMS algorithm in the particular case where $\kappa = 1$ (PNLMS algorithm). As a result, we can expect the two algorithms (IPNLMS and EG±) to have similar performance. On the other hand, if $u \gg \|\mathbf{h}_\mathrm{t}\|_1$, it can be shown that $h_l^+(n) + h_l^-(n) \approx u/L$, $\forall l$. In this case, the EG± algorithm will behave like the IPNLMS with $\kappa = -1$ (NLMS algorithm). Thus, the parameter κ in the IPNLMS operates like the parameter u in the EG±. However, the advantage of the IPNLMS is that no *a priori* information of the system impulse response is required in order to have a better convergence rate than the NLMS algorithm. Another clear advantage of the IPNLMS is that it is much less complex to implement than the EG±. We conclude that IPNLMS is a good approximation of EG± and is more useful in practice. Note also that the approximated EG± algorithm (36) belongs to the family of natural gradient algorithms [9], [10].

5. SIMULATIONS

In this section, we compare by way of simulation, the different algorithms derived in the previous sections. The experiment considers the identification of a single-channel system. The system impulse response $\mathbf{h}_\mathrm{t}$ to be identified is sparse and is of length $L = 2048$. The same length is used for all the adaptive filters $\mathbf{h}(n)$. The sampling rate is 8 kHz and a white noise signal with 30 dB SNR (signal-to-noise ratio) is added to the output $y(n)$. The input signal $x(n)$ is a white Gaussian signal.

Figures 1 shows the convergence of the normalized misalignment, $\|\mathbf{h}_\mathrm{t} - \mathbf{h}(n)\|_2/\|\mathbf{h}_\mathrm{t}\|_2$, for all the algorithms. In this figure, we compare the NLMS, IPNLMS, and EG± algorithms. Clearly, the IPNLMS and EG± algorithms converge much faster than the NLMS algorithm, while the IPNLMS and EG± show similar performance. Figures 1 also compare the algorithms in a tracking situation when after 3 seconds the sparse impulse response is shifted to the right by 50 samples. According to this simulation, the IPNLMS and EG± algorithms track much better than the NLMS algorithm.

6. CONCLUSION

It seems possible to exploit sparsity in adaptive algorithms. One of the first algorithms to do so is the PNLMS proposed by Duttweiler in [5]. The PNLMS algorithm was introduced in the context of network echo cancellation where there is a strong need to improve convergence rate and tracking. It was known for a long time that unknown echo paths in the network are most of the time sparse and there are many different intuitions on how one should take advantage of that. Kivinen and Warmuth [3] derived the EG± algorithm in the context of computational learning theory. We have shown here that a good approximation of the EG± leads to the PNLMS. As a result, the two algorithms have very similar performance in all the simulations we have done. We have also shown some links between the EG algorithms and LMS, so that with appropriate choice of some parameters, the different algorithms can be identical.

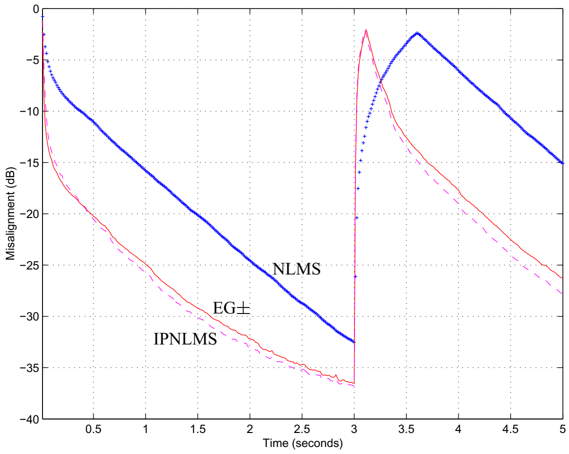

Figure 1: Misalignment of the NLMS (++), IPNLMS (−−), and EG± (−) algorithms with white Gaussian noise as input signal. The impulse response changes at time 3 seconds.

REFERENCES

[1] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Prentice-Hall, Englewood Cliffs, N.J., 1985.

[2] S. Haykin, *Adaptive Filter Theory*. Fourth Edition, Prentice Hall, Upper Saddle River, N.J., 2002.

[3] J. Kivinen and M. K. Warmuth, "Exponentiated gradient versus gradient descent for linear predictors," *Inform. Comput.*, vol. 132, pp. 1–64, Jan. 1997.

[4] J. Benesty, T. Gänsler, D. R. Morgan, M. M. Sondhi, and S. L. Gay, *Advances in Network and Acoustic Echo cancellation*. Springer-Verlag, Berlin, 2001.

[5] D. L. Duttweiler, "Proportionate normalized least mean square adaptation in echo cancelers," *IEEE Trans. Speech Audio Processing*, vol. 8, pp. 508–518, Sept. 2000.

[6] S. I. Hill and R. C. Williamson, "Convergence of exponentiated gradient algorithms," *IEEE Trans. Signal Processing*, vol. 49, pp. 1208–1215, June 2001.

[7] S. Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, pp. 251–276, Feb. 1998.

[8] J. Benesty and S. L. Gay, "An improved PNLMS algorithm," in *Proc. IEEE ICASSP*, 2002.

[9] R. K. Martin, W. A. Sethares, R. C. Williamson, and C. R. Johnson, Jr., "Exploiting sparsity in adaptive filters," in *Conference on Information Sciences and Systems*, The John Hopkins University, 2001.

[10] S. L. Gay and S. C. Douglas, "Normalized natural gradient adaptive filtering for sparse and nonsparse systems," in *Proc. IEEE ICASSP*, 2002.