A Variable Leaky LMS Adaptive Algorithm

Max Kamenetsky and Bernard Widrow
ISL, Department of Electrical Engineering, Stanford University, Stanford CA, USA
e-mail: {maxk,widrow}@stanford.edu
phone: +1 (650) 723-4769
fax: +1 (650) 723-8473

***Abstract*—The LMS algorithm has found wide application in many areas of adaptive signal processing and control. We introduce a variable leaky LMS algorithm, designed to overcome the slow convergence of standard LMS in cases of high input eigenvalue spread. The algorithm uses a greedy punish/reward heuristic together with a quantized leak adjustment function to vary the leak. Simulation results confirm that the new algorithm can significantly outperform standard LMS when the input eigenvalue spread is high.**

I. Introduction

Consider the linear estimation system shown in Fig. 1, where the input $\boldsymbol{x}_k \in \mathcal{R}^N$ is a stationary zero-mean vector random process with autocorrelation matrix $\boldsymbol{R} \stackrel{\triangle}{=} E\left[\boldsymbol{x}_k \boldsymbol{x}_k^T\right]$ for all k, the desired output d_k is a stationary zero-mean scalar random process, $\boldsymbol{w}_k \in \mathcal{R}^N$ is the weight vector, and k is the time index. The system output at time k is given by $y_k = \boldsymbol{w}_k^T \boldsymbol{x}_k$, and the error ϵ_k is computed via $\epsilon_k = d_k - y_k$. Assume that $\boldsymbol{x}_k$ and d_k are jointly stationary with crosscorrelation vector $\boldsymbol{p} \stackrel{\triangle}{=} E[d_k \boldsymbol{x}_k]$ for all k. Define a convex cost function

$$\xi \stackrel{\triangle}{=} E\left[\epsilon_k^2\right] = E\left[d_k^2\right] - 2\boldsymbol{p}^T \boldsymbol{w} + \boldsymbol{w}^T \boldsymbol{R} \boldsymbol{w}.$$

This cost function is the mean square error (MSE). It can easily be shown that, if $\boldsymbol{R}$ is full rank, the unique optimal fixed weight vector which minimizes ξ is given by

$$\boldsymbol{w}^* = \boldsymbol{R}^{-1} \boldsymbol{p}. \tag{1}$$

This is called the Wiener solution [1]. The MSE when using $\boldsymbol{w}^*$ is denoted by ξ^*.

The LMS algorithm of Widrow and Hoff [2] is an iterative algorithm which can be used to compute $\boldsymbol{w}^*$ when the statistics $\boldsymbol{R}$ and $\boldsymbol{p}$ are unknown. Using an instantaneous squared error $\hat{\xi}_k = \epsilon_k^2$ that is quadratic in the weight vector $\boldsymbol{w}_k$, the algorithm uses gradient descent to find the optimal Wiener solution. Accordingly, given $\boldsymbol{w}_k$, the algorithm's weight update equation to compute $\boldsymbol{w}_{k+1}$, the weight vector at the next iteration, is given by

$$\begin{aligned} \boldsymbol{w}_{k+1} &= \boldsymbol{w}_k - \mu \frac{\partial \hat{\xi}_k}{\partial \boldsymbol{w}_k} \\ &= \boldsymbol{w}_k + 2\mu \epsilon_k \boldsymbol{x}_k, \end{aligned} \tag{2}$$

where μ is a user-selectable step size parameter. It has been shown [3] that, for the case of uncorrelated Gaussian data, $\boldsymbol{w}_k \xrightarrow{\text{in m.s.}} \boldsymbol{w}^*$ as $k \to \infty$ if $0 < \mu \leq \frac{1}{3\,\mathrm{Tr}(\boldsymbol{R})}$.

In steady-state, the weight vector $\boldsymbol{w}_k$ undergoes Brownian motion around the Wiener solution $\boldsymbol{w}^*$. Consequently, there is an excess MSE even as $k \to \infty$. A good approximation for this excess MSE is given in [4] by $\xi_{\text{excess}} = \mu \xi^* \,\mathrm{Tr}(\boldsymbol{R})$.

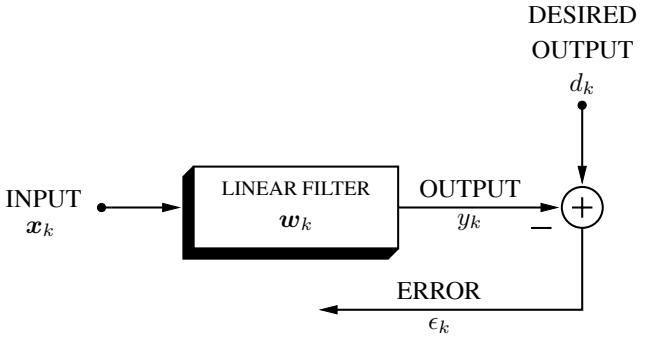

Fig. 1. Linear estimation problem.

During the transient phase, each of the N weight vector elements exponentially relaxes to its optimal value with a time constant of $\tau_n \approx \frac{1}{2\mu\lambda_n}$, where λ_n is the nth eigenvalue of $\boldsymbol{R}$. For example, consider the contours of constant MSE shown in Fig. 2. We start the algorithm at three locations in the weight-vector space, with all three locations corresponding to the same initial MSE. We then proceed to run the LMS algorithm for 11 iterations and plot the resulting weight vector values. It can be seen that the algorithm converges much slower along the worst-case eigendirection (the direction of the eigenvector corresponding to the smallest eigenvalue of $\boldsymbol{R}$) as opposed to the best-case eigendirection (the direction of the eigenvector corresponding to the largest eigenvalue of $\boldsymbol{R}$). This disparity increases as the eigenvalue spread $\lambda_{\text{max}}/\lambda_{\text{min}}$ increases – corresponding to an equivalent increase in the eccentricity of the elliptical contours of constant MSE [4]. Thus, the key step in improving the transient performance of LMS lies in decreasing the input eigenvalue spread.

Now consider a new instantaneous cost function that places a penalty on the norm of the weight vector:

$$J_k = \epsilon_k^2 + \gamma \boldsymbol{w}_k^T \boldsymbol{w}_k, \tag{3}$$

where $\gamma \geq 0$ is a user-selectable parameter. The term $\gamma \boldsymbol{w}_k^T \boldsymbol{w}_k$ can be viewed as a regularization parameter or an effort penalty [5]. A recursive algorithm for minimizing the cost function in (3) can be derived via

$$\begin{aligned} \boldsymbol{w}_{k+1} &= \boldsymbol{w}_k - \mu \frac{\partial J_k}{\partial \boldsymbol{w}_k} \\ &= (1 - 2\mu\gamma)\boldsymbol{w}_k + 2\mu \epsilon_k \boldsymbol{x}_k. \end{aligned} \tag{4}$$

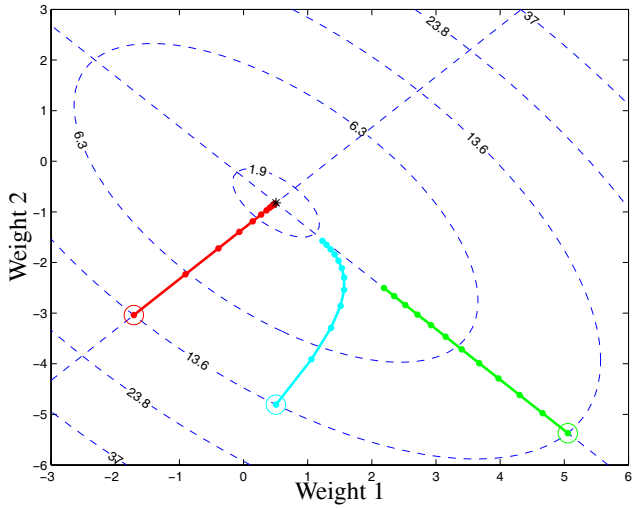

Fig. 2. Convergence of LMS-adapted weight vectors.

This algorithm is known as the leaky LMS algorithm, and the parameter γ is referred to as the "leak." The name stems from the fact that, when the input is turned off, the weight vector of the regular LMS algorithm stalls. With leaky LMS in the same scenario, the weight vector instead "leaks" out. It has been shown that leaky LMS can be used to improve stability in a finite-precision implementation [6], ameliorate the effects of nonpersistent excitation [7], and reduce undesirable effects like stalling [8], bursting [9], etc. Detailed analysis of the performance of this algorithm is given in [10] and [11]. In particular, it can be shown that

$$\lim_{k\to\infty} E[\boldsymbol{w}_k] = (\boldsymbol{R}+\gamma\boldsymbol{I})^{-1}\boldsymbol{p}, \tag{5}$$

and thus the leaky LMS algorithm can be interpreted as adding zero-mean white noise with autocorrelation matrix $\gamma\boldsymbol{I}$ to the input $\boldsymbol{x}$.

II. Variable Leaky LMS Algorithm

Since $\lim_{k\to\infty} E[\boldsymbol{w}_k] \neq \boldsymbol{w}^*$, the leaky algorithm is biased. However, it can also be seen that, by adding white noise to the input, the leaky algorithm effectively decreases the input eigenvalue spread. That is, if $\{\lambda_1,\ldots,\lambda_N\}$ are the input eigenvalues "seen" by the standard LMS algorithm, then $\{\lambda_1+\gamma,\ldots,\lambda_N+\gamma\}$ are the eigenvalues "seen" by the leaky LMS algorithm. Thus, since $\gamma \geq 0$, the new eigenvalue spread is smaller than the original eigenvalue spread:

$$\frac{\lambda_{\max}+\gamma}{\lambda_{\min}+\gamma} \leq \frac{\lambda_{\max}}{\lambda_{\min}}. \tag{6}$$

This means that the leaky algorithm's worst-case transient performance will be better than that of the standard LMS algorithm.

On the one hand, we would like γ to be as large as possible (while still making sure the algorithm converges) in order to achieve the greatest possible reduction in eigenvalue spread. On the other hand, we would like γ to be as small as possible because any non-zero γ biases the solution. In order to facilitate the tradeoff between the two objectives, we now introduce a variable leaky LMS (VL-LMS) algorithm:

$$\boldsymbol{w}_{k+1} = (1-2\mu\gamma_k)\boldsymbol{w}_k + 2\mu\epsilon_k\boldsymbol{x}_k, \tag{7}$$

where γ_k is now a time-varying parameter. Note that the LMS algorithm is a special case of VL-LMS when $\gamma_k = 0$. In a stationary environment, we would like the leak γ_k to be large in the transient phase in order to speed up convergence. As we reach steady-state, we would like to gradually decrease the leak. Of course, this procedure also needs to work in nonstationary environments, so the leak adjustment must be adaptive.

We need to answer two questions in order to design an adaptive leak adjustment algorithm. The first question is when to adjust the leak, i.e. when should it be increased and when should it be decreased? The second question is how much to adjust the leak. Let us deal with these questions one by one.

In order to answer the question of when to adjust the leak, consider the *a posteriori* LMS error $\tilde{\epsilon}_k^{\text{LMS}}$:

$$\begin{aligned}\tilde{\epsilon}_k^{\text{LMS}} &\triangleq d_k - \boldsymbol{w}_{k+1}^T\boldsymbol{x}_k \\ &= d_k - (\boldsymbol{w}_k^T + 2\mu\epsilon_k\boldsymbol{x}_k^T)\boldsymbol{x}_k \\ &= \epsilon_k(1-2\mu\boldsymbol{x}_k^T\boldsymbol{x}_k).\end{aligned} \tag{8}$$

It can be seen that this is the error when using the new weight vector $\boldsymbol{w}_{k+1}$ together with the old desired response d_k and input vector $\boldsymbol{x}_k$. Similarly, consider the *a posteriori* VL-LMS error $\tilde{\epsilon}_k^{\text{VL-LMS}}$ when using the current value of the leak:

$$\begin{aligned}\tilde{\epsilon}_k^{\text{VL-LMS}} &\triangleq d_k - \boldsymbol{w}_{k+1}^T\boldsymbol{x}_k \\ &= d_k - ((1-2\mu\gamma_k)\boldsymbol{w}_k^T + 2\mu\epsilon_k\boldsymbol{x}_k^T)\boldsymbol{x}_k \\ &= d_k - (1-2\mu\gamma_k)\boldsymbol{w}_k^T\boldsymbol{x}_k - 2\mu\epsilon_k\boldsymbol{x}_k^T\boldsymbol{x}_k.\end{aligned} \tag{9}$$

Finally, consider the following simple leak adjustment algorithm, to be performed with each iteration. If $|\tilde{\epsilon}_k^{\text{VL-LMS}}| < |\tilde{\epsilon}_k^{\text{LMS}}|$, then we will increase the leak. Otherwise, we will decrease the leak. This leak adjustment scheme is based on a greedy punish/reward heuristic that increases the leak when VL-LMS would outperform LMS and decreases the leak when LMS would outperform VL-LMS.

The second question we need to answer is by how much to increase or decrease the leak. When answering this question, we need to keep the following two considerations in mind. First, we need to have the ability to quickly vary the leak if the leak is large. The rationale for this is that continuing with a large leak when $\tilde{\epsilon}_k^{\text{LMS}} \ll \tilde{\epsilon}_k^{\text{VL-LMS}}$ can have a large adverse impact on the convergence speed of this algorithm. Second, we need to be able to slowly vary the leak if the leak is small. The rationale for this is that a small leak will frequently correspond to cases when the algorithm is close to steady-state, and large jumps in the leak due to gradient noise will adversely affect steady-state performance.

Keeping the above two considerations in mind, here is one simple procedure for determining the next leak γ_{k+1} given the current leak γ_k. Consider the following quantized exponential function:

$$\gamma = f(m) = \gamma_{\max}(m/M)^{\alpha}, \tag{10}$$

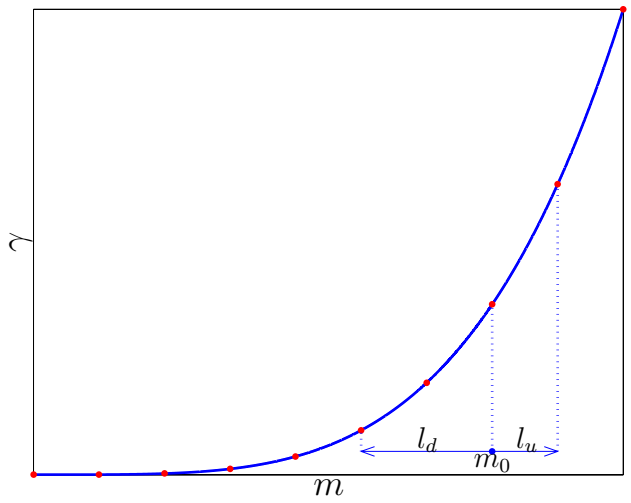

Fig. 3. Quantized leak adjustment function.

TABLE I
VL-LMS LEAK ADJUSTMENT ALGORITHM.

```
set m = 1
while k is being incremented
  if |ē_k^VL-LMS| < |ē_k^LMS|
    set m = min(m + l_u, M)
  else
    set m = max(m − l_d, 0)
  endif
endwhile
```

where $m = 0, \ldots, M$ is the independent variable, and $M \in \mathcal{Z}$, $\alpha \in \mathcal{R}$, and $\gamma_{\max} \in \mathcal{R}$ are positive user-defined parameters. An example of this function is shown in Fig. 3. Consider the case where we are at location m_0, corresponding to leak $\gamma = f(m_0)$, at the current iteration. Then, if we decide to increase the leak, we will increase m_0 by a user-defined constant $l_u \in \mathcal{Z}$ and use the new leak $\gamma = f(m_0 + l_u)$. If we decide to decrease the leak, we will decrease m_0 by a user-defined constant $l_d \in \mathcal{Z}$ and use the new leak $\gamma = f(m_0 - l_u)$. This is shown graphically in Fig. 3. The complete pseudo-code for this algorithm is shown in Table I. Note that the algorithm starts out with $m = 1$, which corresponds to a small value of the leak.

III. Simulation Results

We now present some simulation results to demonstrate the performance of the VL-LMS algorithm. In all cases, we will consider a system identification problem where we are trying to identify the weights of a 12-tap FIR lowpass plant with cutoff frequency at 0.6π radians. The problem setup is shown in Fig. 4 where $\boldsymbol{w}^*$ corresponds to the weights of the plant. The output noise n_k is IID, zero-mean, and independent of $\boldsymbol{x}_k$. The output noise variance ξ^* (which also corresponds to the minimum MSE) is -15 dB. The VL-LMS algorithm parameters are set to $M = 200$, $\alpha = 4$, $l_u = 1$, and $l_d = 3$. The step-size parameter μ is adjusted for an excess MSE of $1.05\xi^*$. All of the learning curves are averaged over 200 runs.

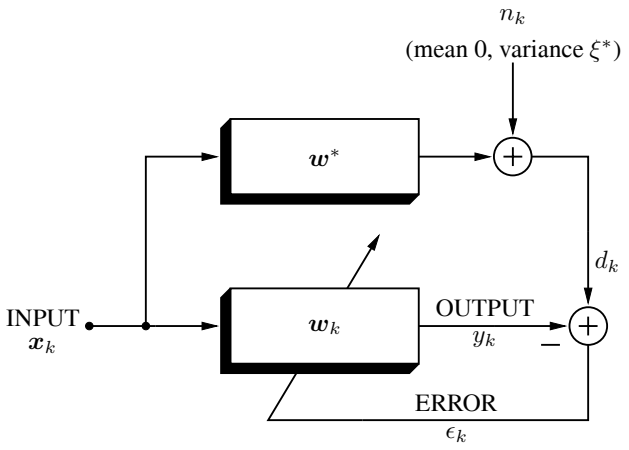

Fig. 4. Adaptive system identification.

A. Simulation 1

In the first simulation, the input is a first-order Markov process obtained by passing white Gaussian noise through an autoregressive filter with impulse response

$$h[k] = \rho^k u[k], \tag{11}$$

where $u[k]$ is a step function at time $k = 0$. The Markov parameter ρ is set to 0.999, so the resulting input eigenvalue spread is rather high ($\lambda_{\max}/\lambda_{\min} \approx 2.3 \times 10^4$).

Fig. 5 shows the convergence of LMS and VL-LMS when started along the worst-case eigenvector. As expected, because of the high eigenvalue spread, LMS converges rather slowly. On the other hand, VL-LMS converges much faster when compared with LMS.

Fig. 6 shows the convergence of LMS and VL-LMS when started along the best-case eigenvector. LMS converges very fast when started from this direction and cannot be outperformed by the VL-LMS algorithm. Therefore, VL-LMS attempts to keep the leak as small as possible and follow the LMS trajectory as best as it can. Consequently, the trajectories of the two algorithms virtually overlap.

B. Simulation 2

In the second simulation, the input is a discrete-time sinusoid with radian frequency $\omega_0 = 2\pi/3$ and additive white Gaussian noise. The input SNR is 30 dB, and consequently, the eigenvalue spread of the input autocorrelation matrix is once again rather high ($\lambda_{\max}/\lambda_{\min} \approx 6 \times 10^3$).

Fig. 7 shows the convergence of LMS and VL-LMS when started along the worst-case eigenvector. Once again, because of the high eigenvalue spread, LMS converges rather slowly. On the other hand, VL-LMS converges much faster when compared with LMS.

Fig. 8 shows the convergence of LMS and VL-LMS when started along the best-case eigenvector. Once again, VL-LMS cannot outperform LMS when started along this eigenvector, and VL-LMS thus once again attempts to keep the leak as small as possible and follow the LMS trajectory as best as it can. In this plot, the two trajectories are the same to within a pixel.

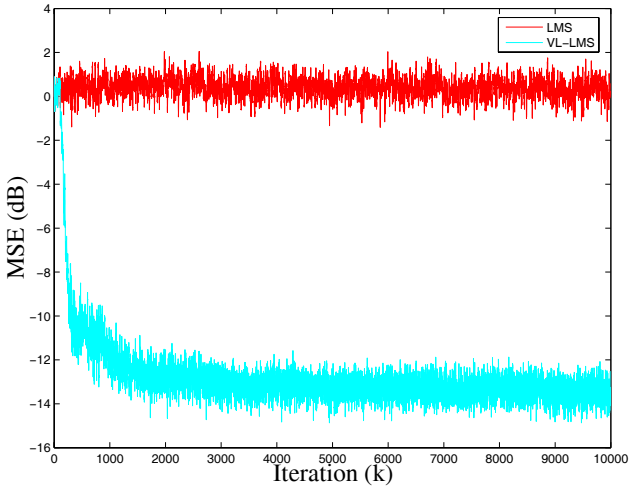

Fig. 5. Simulation 1: Worst-case eigendirection.

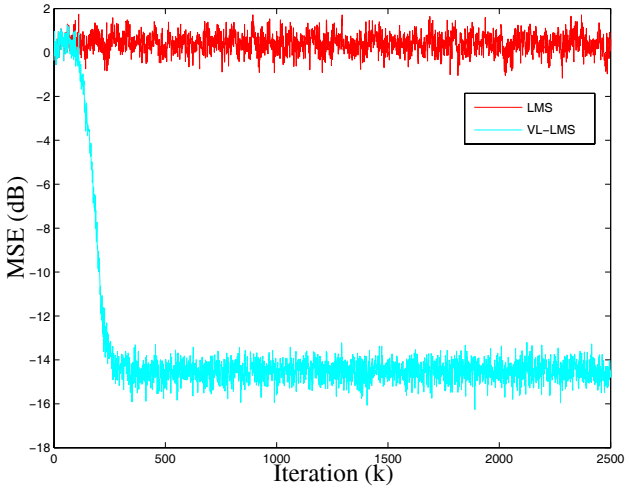

Fig. 7. Simulation 2: Worst-case eigendirection.

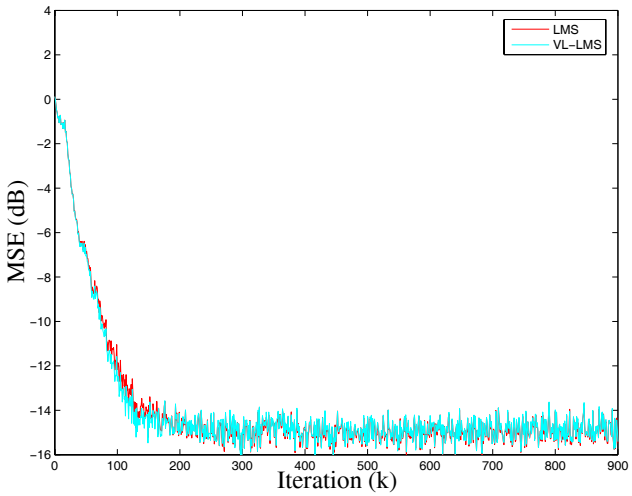

Fig. 6. Simulation 1: Best-case eigendirection.

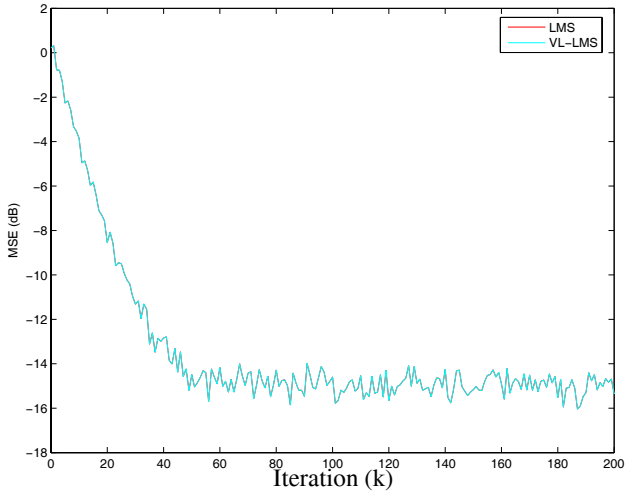

Fig. 8. Simulation 2: Best-case eigendirection.

IV. Conclusion

We described a new variable leaky LMS (VL-LMS) adaptive algorithm with an adaptive leak parameter γ_k. We created a leak adjustment algorithm for γ_k that facilitates a tradeoff between reducing eigenvalue spread in the transient phase and keeping the bias small in steady-state. Simulation results show that the VL-LMS algorithm can significantly outperform the standard LMS algorithm.

References

[1] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*, Prentice Hall, Upper Saddle River, NJ, 2000.

[2] B. Widrow and M. E. Hoff, "Adaptive switching circuits," in *IRE WESCON Conv. Rec.*, 1960, vol. 4, pp. 96–104.

[3] A. Feuer and E. Weinstein, "Convergence analysis of LMS filters with uncorrelated Gaussian data," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-33, no. 1, pp. 222–229, Feb. 1985.

[4] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Prentice Hall, Upper Saddle River, NJ, 1985.

[5] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer-Verlag, 2001.

[6] J. M. Cioffi, "Limited-precision effects in adaptive filtering," *IEEE Trans. Circuits Syst.*, vol. CAS-34, no. 7, pp. 821–833, July 1987.

[7] W. A. Sethares, D. A. Lawrence, C. R. Johnson, Jr., and R. R. Bitmead, "Parameter drift in LMS adaptive filters," *IEEE Trans. Acoust. Speech Signal Process.*, vol. ASSP-34, no. 4, pp. 868–879, Aug. 1986.

[8] A. Segalen and G. Demoment, "Constrained LMS adaptive algorithm," *Electron. Lett.*, vol. 18, no. 5, pp. 226–227, Mar. 4 1982.

[9] G. J. Rey, R. R. Bitmead, and C. R. Johnson, Jr., "The dynamics of bursting in simple adaptive feedback systems with leakage," *IEEE Trans. Circuits Syst.*, vol. 38, no. 5, pp. 476–488, May 1991.

[10] K. Mayyas and T. Aboulnasr, "Leaky LMS algorithm: MSE analysis for Gaussian data," *IEEE Trans. Signal Process.*, vol. 45, no. 4, pp. 927–934, Apr. 1997.

[11] A. H. Sayed and T. Y. Al-Naffouri, "Mean-square analysis of normalized leaky adaptive filters," in *Proc. ICASSP*, Salt Lake City, UT, May 2001, vol. 6, pp. 3873–3876.