Subspace Leaky LMS

Brian D. Rigling, *Member, IEEE,* and Philip Schniter, *Member, IEEE*

***Abstract*—The least mean squared (LMS) adaptive filtering algorithm may experience uncontrolled parameter drift when its input signal is not persistently exciting, leading to serious consequences when implemented with finite word-length. Though so-called "tap-leakage" modifications of LMS have been proposed to mitigate this drift, they inevitably introduce parameter bias which degrades mean-squared error performance. In this letter, we propose a novel algorithm which leaks only in the unexcited modes, thus introducing insignificant bias, while still retaining the low computational complexity of LMS.**

***Index Terms*—Adaptive filtering, leakage, leaky least mean squares, least mean squares (LMS), subspace tracking.**

I. Introduction

THE LEAST mean squared (LMS) algorithm [1]–[3] is a method of linear adaptive filtering belonging to the family of stochastic gradient algorithms. LMS can be described by the two-step filter-parameter update procedure

$$e(n) = d(n) - \underline{w}^H(n)\underline{u}(n) \quad (1)$$

$$\underline{w}(n+1) = \underline{w}(n) + \mu\underline{u}(n)e^*(n) \quad (2)$$

where $d(n)$ denotes the reference signal, $e(n)$ the error signal, $\underline{u}(n)$ the vector-valued input signal at time $n, \underline{w}(n)$ the filter parameters, and μ a small positive step-size. Essentially, LMS attempts to adjust the filter parameters $\underline{w}(n)$ so that the mean-squared error (MSE) $\mathrm{E}\{|e(n)|^2\}$ is minimized. Note from (2) that the update of $\underline{w}(n)$ is restricted to the space of vectors spanned by the previous inputs $\{\underline{u}(i) : i < n\}$, implying that it is impossible for LMS to adjust any components of $\underline{w}(n)$ that are not active in the input signal.

Motivated by [4], we consider the following partition of the input signal space. Components of $\underline{u}(n)$ that consistently contain energy are said to belong to the *persistently excited* (PE) subspace, while components of $\underline{u}(n)$ that consistently contain no energy are said to belong to the *unexcited* (UE) subspace. If LMS is implemented on a finite-precision processor, quantization noise can cause uncontrolled parameter drift in the UE subspace. If the processor has finite word length, parameter drift may cause numerical overflow that can lead to system instability.

The *leaky LMS* (L-LMS) algorithm [3], [5] has been widely used to prevent filter tap drift and maintain stability with inputs that are not PE in all modes. L-LMS is a modification of standard LMS (1)–(2) in which (2) is replaced by the leaky update $\underline{w}(n+1) = (1-\mu\gamma)\underline{w}(n) + \mu\underline{u}(n)e^*(n)$. Through proper choice of positive leakage parameter γ, excess parameter drift can be avoided. Though L-LMS has very low implementation complexity, it applies filter tap leakage indiscriminately, thus biasing $\underline{w}(n)$ and increasing MSE.

Circular leaky LMS (CL-LMS) [7] "leaks" filter taps only when tap drift is in danger of causing numerical overflow. CL-LMS examines one parameter per iteration and attenuates that parameter only if it exceeds a specified threshold. The computational complexity of CL-LMS is only slightly greater than that of standard LMS. The application of leakage, however, still introduces parameter bias and therefore increases MSE.

In this letter, we propose a *subspace leaky LMS* (SL-LMS) algorithm that periodically applies leakage that is significant in the UE subspace, while insignificant in the PE subspace. Selective application of leakage is accomplished by tracking the UE subspace. While the computational cost of SL-LMS is greater than that of the previously mentioned algorithms, the complexity is still linear in the number of filter parameters.

The following notation is used. Column vectors are denoted by underlined lowercase letters, matrices are denoted by uppercase letters. Vector elements are denoted using subscripts (e.g., x_m), while time is indexed in parentheses (e.g., $\underline{x}(n)$). The conjugate transpose of a vector or matrix is denoted by the superscript $(\cdot)^H$ and complex conjugation is denoted by $(\cdot)^*$. Finally, I denotes the identity matrix, $\delta(\cdot)$ the Kronecker delta, and $\langle\cdot\rangle_M$ the modulo-M operation.

II. Background

The signals $\underline{u}(n)$ and $d(n)$ are assumed to be wide-sense stationary (WSS) and possibly nonzero mean. Eigendecomposition of $R_u = \mathrm{E}\{\underline{u}(n)\underline{u}^H(n)\}$ leads to

$$R_u = V\Lambda V^H = [V_1 \quad V_2]\begin{bmatrix}\Lambda_1 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}V_1^H\\ V_2^H\end{bmatrix} \quad (3)$$

where the columns of V_1 span the PE subspace, $\mathcal{R}\{V_1\}$, and correspond to the nonzero eigenvalues in the diagonal matrix Λ_1. The columns of V_2 correspond to zero-valued eigenvalues and span the UE subspace $\mathcal{R}\{V_2\}$.

A. LMS

The LMS algorithm [1]–[3] adapts $\underline{w}(n) \in \mathbb{C}^M$ in an attempt to minimize the MSE cost $J(n) = \mathrm{E}\{|e(n)|^2\}$. Gradient-descent minimization of $J(n)$ yields $\underline{w}(n+1) = \underline{w}(n) + \mu\mathrm{E}\{\underline{u}(n)e^*(n)\}$, and LMS approximates this recursion using (2). Under LMS, $\{\underline{w}(n)\}$ converges in mean to the MSE-minimizing Wiener solution, i.e., $R_u\underline{w} = \mathrm{E}\{\underline{u}(n)d^*(n)\}$ [3]. If R_u is not full rank, the Wiener solution is not unique.

Manuscript received March 13, 2003; revised April 23, 2003. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yiten (Arden) Huang.

The authors are with the Department of Electrical Engineering, The Ohio State University, Columbus, OH 43210 USA (e-mail: riglingb@ee.eng.ohio-state.edu; schniter@ee.eng.ohio-state.edu).

Digital Object Identifier 10.1109/LSP.2003.821760

In a departure from the typical LMS analysis, we assume a *noisy* parameter update of the form

$$\underline{w}(n+1) = \underline{w}(n) + \mu\underline{u}(n)e^*(n) + \underline{q}(n) \quad (4)$$

where a (possibly nonzero mean) disturbance $\underline{q}(n)$, perhaps due to finite-precision effects, is added at each iteration.

Using (3), we can break the noisy LMS update (4) into two decoupled equations, the first corresponding to the PE subspace and the second to the UE subspace

$$V_1^H\underline{w}(n+1) = V_1^H\underline{w}(n) + \mu V_1^H\underline{u}(n)e^*(n) + V_1^H\underline{q}(n) \quad (5)$$

$$V_2^H\underline{w}(n+1) = V_2^H\underline{w}(n) + V_2^H\underline{q}(n). \quad (6)$$

The components of $\underline{q}(n)$ that lie in the PE subspace $\mathcal{R}\{V_1\}$ increase MSE; the components of $\underline{q}(n)$ that lie in the UE subspace do not increase MSE, but cause $\underline{w}(n)$ to undergo a parameter drift within $\mathcal{R}\{V_2\}$. Our goal is to simultaneously minimize MSE $\mathrm{E}\{|e(n)|^2\}$ and drift of $\underline{w}(n)$ within $\mathcal{R}\{V_2\}$. Thus, we desire that $\underline{w}(n)$ equals the minimum-norm version of the (MSE-minimizing, possibly nonunique) Wiener solution, i.e., $\underline{w}_\star = V_1\Lambda_1^{-1}V_1^H\mathrm{E}\{\underline{u}(n)d^*(n)\}$.

B. Leaky LMS

The L-LMS algorithm attempts to control the drift of $\underline{w}(n)$ by adding a term to the adaptation cost function that penalizes the size of $\underline{w}(n)$

$$\mathrm{J}_{\mathsf{l}}(n) = \mathrm{E}\{|e(n)|^2\} + \gamma\|\underline{w}(n)\|^2. \quad (7)$$

This leads to the (noisy) L-LMS stochastic gradient update

$$\underline{w}(n+1) = (1-\mu\gamma)\underline{w}(n) + \mu\underline{u}(n)e^*(n) + \underline{q}(n) \quad (8)$$

where the positive parameter γ determines the degree of penalty placed on parameter drift. A larger γ reduces parameter drift within the UE subspace but increases parameter bias (leading to increased MSE). These and other properties of L-LMS are described in [3], [6], and [8].

C. Circular Leaky LMS

In contrast to L-LMS, the CL-LMS algorithm [7] discriminately applies leakage to a single coefficient of $\underline{w}(n)$ per iteration. The cost function for CL-LMS may be written as

$$\mathrm{J}_{\mathsf{C}}(n) = \mathrm{E}\{|e(n)|^2\} + \alpha(n)|w_{\langle n\rangle_M}(n)|^2 \quad (9)$$

where

$$\alpha(n) = \begin{cases} \alpha_0, & \text{if } |w_{\langle n\rangle_M}(n)| \geq C_2 \\ \alpha_0 - \frac{\alpha_0}{2}\left(\frac{C_2 - |w_{\langle n\rangle_M}(n)|}{D}\right)^2, & \text{if } C_1 + D \leq |w_{\langle n\rangle_M}(n)| < C_2 \\ \frac{\alpha_0}{2}\left(\frac{|w_{\langle n\rangle_M}(n)| - C_1}{D}\right)^2, & \text{if } C_1 < |w_{\langle n\rangle_M}(n)| < C_1 + D \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

Appropriate choices for the CL-LMS parameters $\{\alpha_0, C_1, C_2, D\}$ are described in [7]. This CL-LMS cost function leads to the (noisy) update equation

$$\underline{w}(n+1) = \underline{w}(n) - \mu\alpha(n)\underline{\delta}_{\langle n\rangle_M}w_{\langle n\rangle_M}(n) + \mu\underline{u}(n)e^*(n) + \underline{q}(n) \quad (11)$$

where $\underline{\delta}_k$ is a column vector consisting of zeros except for a one in the kth position. As with L-LMS, CL-LMS reduces drift through tap-leakage. However, because leakage is applied discriminately, bias and MSE are decreased relative to L-LMS. In [7] it is noted that, when all modes of the system are PE, the CL-LMS algorithm yields unbiased parameter estimates. In this latter case, however, tap-leakage is not necessary and standard LMS is preferable due to its low complexity.

III. Subspace Leaky LMS

As with CL-LMS, the SL-LMS algorithm seeks to discriminately apply leakage. Unlike CL-LMS, though, SL-LMS tracks the UE subspace of the input signal and applies leakage only within that subspace.

A. Algorithm

The SL-LMS algorithm consists of two parts: 1) tracking the UE subspace; and 2) updating $\underline{w}(n)$. To track the UE subspace, we seek $\underline{s}(n) \in \mathbb{C}^M$ that minimizes the cost

$$\mathrm{J}_{\mathsf{SS}}(n) = \mathrm{E}\{|\underline{s}^H(n)\underline{u}(n)|^2\}. \quad (12)$$

The following lemma states the desirable properties of $\mathrm{J}_{\mathsf{SS}}(n)$.

Lemma 1: If $\underline{s}(n)$ is nonzero, then $\underline{s}(n)$ minimizes $\mathrm{J}_{\mathsf{SS}}(n)$ if and only if $\underline{s}(n)$ lies within the null space of R_u.

Proof: Clearly, we have $\mathrm{J}_{\mathsf{SS}}(n) = \underline{s}^H(n)R_u\underline{s}(n)$. Thus, for nonzero $\underline{s}(n)$, $\mathrm{J}_{\mathsf{SS}}(n) = 0$ if and only if $\underline{s}(n) \in \mathcal{R}\{V_2\}$. □

The gradient descent minimization of $\mathrm{J}_{\mathsf{SS}}(n)$ can be approximated by the (noisy) stochastic gradient update

$$\underline{s}(n+1) = \underline{s}(n) - \zeta(n)\underline{u}(n)\underline{u}^H(n)\underline{s}(n) + \underline{q}'(n) \quad (13)$$

where $\zeta(n)$ is a time-varying step-size and $\underline{q}'(n)$ models finite-precision effects in the tracking vector update. It is important to note that, with mild assumptions on $\zeta(n)$ and $\underline{q}'(n)$, tracking vector $\underline{s}(n)$ will converge to zero if initialized within the PE subspace. Setting $\zeta(n) = (\underline{u}^H(n)\underline{u}(n))^{-1}$ in the fashion of normalized-LMS [3] accelerates convergence and gives

$$\underline{s}(n+1) = \underline{s}(n) - c(n)\underline{u}(n) + \underline{q}'(n) \quad (14)$$

$$c(n) = (\underline{u}^H(n)\underline{u}(n))^{-1}\underline{u}^H(n)\underline{s}(n). \quad (15)$$

In SL-LMS we initialize the tracking vector $\underline{s}(0)$ to the first member of $\mathcal{S} = \{\underline{s}^{(0)}, \ldots \underline{s}^{(M-1)}\}$, an orthonormal basis for $\mathbb{C}^M$. After N iterations, $\underline{s}(n)$ is reinitialized to the next element in $\mathcal{S}$, and adaptation of the tracking vector begins again. After MN iterations, all elements of $\mathcal{S}$ have been used as reinitializations, and the process is repeated from the start.

Using the previously described method of subspace tracking, we update the parameters $\underline{w}(n)$ according to stochastic gradient descent of the SL-LMS cost

$$\mathrm{J}_{\mathsf{Sl}}(n) = \mathrm{E}\{|e(n)|^2\} + \delta(\langle n\rangle_N - N + 1)|\underline{s}^H(n)\underline{w}(n)|^2. \quad (16)$$

Thus, a penalty, determined by the degree to which the filter parameters have drifted in the $\underline{s}(n)$ direction, is added to the cost function, but only after $\underline{s}(n)$ has adapted for a specified number of iterations N. The number N should be large enough to allow sufficient convergence of the tracking vector but small enough to prevent significant drift from occurring in $\underline{w}(n)$ and $\underline{s}(n)$. This cost function leads to the following (noisy) SL-LMS update:

$\underline{s}(0) = \underline{s}^{(0)}, \underline{w}(0) = \underline{0},$
for $n = 0, 1, 2, 3 \ldots$
  $c(n) = (\underline{u}^H(n)\underline{s}(n))/(\underline{u}^H(n)\underline{u}(n))$
  if $\langle n \rangle_N = N - 1$
    $\underline{s}(n+1) = \underline{s}(n) - c(n)\underline{u}(n) + \underline{q}'(n)$
    $\underline{w}(n+1) = (I - \underline{s}(n)\underline{s}^H(n))\underline{w}(n) + \mu \underline{u}(n)e^*(n) + \underline{q}(n)$
    $\underline{s}(n+1) = \underline{s}^{\langle \lfloor (n+1/N) \rfloor \rangle_M}$
  else
    $\underline{s}(n+1) = \underline{s}(n) - c(n)\underline{u}(n) + \underline{q}'(n)$
    $\underline{w}(n+1) = \underline{w}(n) + \mu \underline{u}(n)e^*(n) + \underline{q}(n).$

Recall that $\underline{q}(n)$ and $\underline{q}'(n)$ model the effects of finite precision; there is no need to explicitly implement these noises!

B. Mean Convergence of Tracking Vectors

The mean behavior of the subspace tracking vectors lends insight into the mean behavior of SL-LMS. To proceed further it is convenient to transform $\underline{s}(n)$ and $\underline{u}(n)$ into

$$\underline{\eta}(n) = V^H \underline{s}(n) \tag{17}$$

$$\underline{\beta}(n) = V^H \underline{u}(n) \tag{18}$$

so that $\underline{s}(n) = V\underline{\eta}(n)$ and $\underline{u}(n) = V\underline{\beta}(n)$. If we define $\eta_1^H(n) = V_1^H \underline{s}(n)$ and $\eta_2^H(n) = V_2^H \underline{s}(n)$, then $\underline{\eta}_1(n)$ will track the PE subspace, while $\underline{\eta}_2(n)$ will track the UE subspace.

For simplicity, we assume (e.g., as in [3] and [9])

(A1) $\underline{\beta}^H(n)\underline{\beta}(n) = \sigma_u^2$ (e.g., from large M).

(A2) $\underline{s}(n)$ and $\underline{u}(n)$ independent (e.g., from small step-size).

Using (A1) one may express (15) as

$$c(n) = \sigma_u^{-2} \underline{\beta}^H(n)\underline{\eta}(n) \tag{19}$$

yielding the following transform of (14):

$$\underline{\eta}(n+1) = \underline{\eta}(n) - \sigma_u^{-2}\underline{\beta}(n)\underline{\beta}^H(n)\underline{\eta}(n) + V^H \underline{q}'(n). \tag{20}$$

Taking the expectation of (20) and invoking (A2)

$$\mathrm{E}\{\underline{\eta}(n+1)\} = \left(I - \sigma_u^{-2}\Lambda\right)\mathrm{E}\{\underline{\eta}(n)\} + V^H \mathrm{E}\{\underline{q}'(n)\}. \tag{21}$$

Using $\sigma_u^2 = \mathrm{tr}(\Lambda_1)$ and induction, we find that the mean (transformed) tracking components can be expressed as

$$\mathrm{E}\{\underline{\eta}_1(n)\} = (I - (\mathrm{tr}(\Lambda_1))^{-1}\Lambda_1)^{\langle n \rangle_N} V_1^H \underline{s}^{\langle \lfloor \frac{n}{N} \rfloor \rangle_M} + V_1^H \sum_{i=1}^{\langle n \rangle_N} \mathrm{E}\left\{\underline{q}'\left(\left\lfloor \frac{n}{N} \right\rfloor N + i\right)\right\} \tag{22}$$

$$\mathrm{E}\{\underline{\eta}_2(n)\} = V_2^H \underline{s}^{\langle \lfloor \frac{n}{N} \rfloor \rangle_M} + V_2^H \sum_{i=1}^{\langle n \rangle_N} \mathrm{E}\left\{\underline{q}'\left(\left\lfloor \frac{n}{N} \right\rfloor N + i\right)\right\}. \tag{23}$$

If $\mathrm{E}\{\underline{q}'(n)\}$ is negligible, then $\mathrm{E}\{\underline{\eta}_1(n)\} \to \underline{0}$, assuming N is large enough to permit convergence (since $0 < \|I - \mathrm{tr}(\Lambda_1)^{-1}\Lambda_1\| < 1$). In this case, the UE components of the most recent reinitialization $\underline{s}^{\langle \lfloor (n/N) \rfloor \rangle_M}$ are unaffected, while the PE components attenuate at a rate proportional to the relative input signal power in that mode. Since the M most recent reinitializations were taken from the orthonormal basis $\mathcal{S}$, the M most recent converged tracking vectors $\underline{s}(n)$ will span the UE subspace. Thus the drift of $\underline{w}(n)$ within the UE subspace may be eliminated by M leaks of $\underline{w}(n)$ in the directions of these converged tracking vectors.

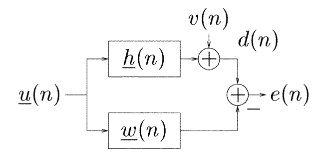

Fig. 1. Parameter identification task.

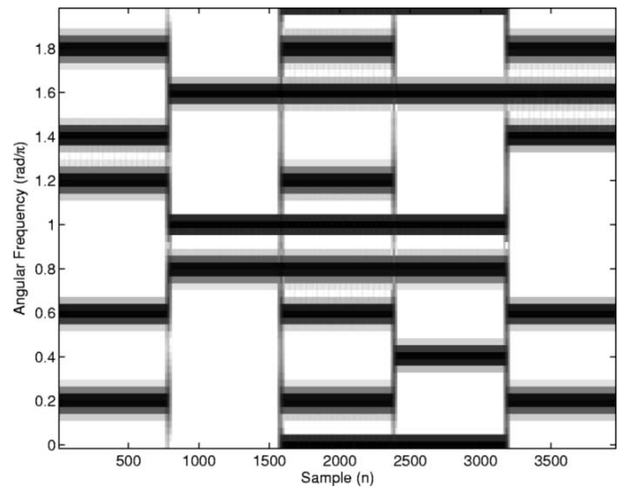

Fig. 2. Spectrogram of the input signal $\underline{u}(n)$.

If $\mathrm{E}\{\underline{q}'(n)\}$ is not negligible, then $\underline{s}(n)$ will drift and we cannot claim that its UE components remain fixed, while its PE components decay to zero. But, since only N updates occur between reinitializations, drift in $\underline{s}(n)$ will be limited. Thus, N should be chosen as a tradeoff between convergence of $\underline{s}(n)$ and drift of $\underline{w}(n)$ and $\underline{s}(n)$. Simulations in Section IV verify that choices of N exist which lead to good behavior.

IV. Simulation Results

The L-LMS, CL-LMS, and SL-LMS algorithms were compared in the parameter identification setting illustrated by Fig. 1, where the objective was to track the ten-tap time-varying response $\{\underline{h}(n)\}$. The signals $\underline{h}(n)$, and $v(n)$ were generated using zero-mean, circular, mutually uncorrelated Gaussian processes constructed such that $\mathrm{E}\{\underline{h}(n)\underline{h}^H(n-k)\} = \delta(k)I$ and $\mathrm{E}\{v(n)v^*(n-k)\} = 0.01\,\delta(k)$. The real and imaginary parts of (nonzero mean) $\underline{q}(n)$ and $\underline{q}'(n)$ were mutually uncorrelated Rayleigh such that $\mathrm{E}\{\underline{q}(n)\underline{q}^H(n-k)\} = \mathrm{E}\{\underline{q}'(n)\underline{q}'^H(n-k)\} = 1 \times 10^{-5}\,\delta(k)I$. The input signal $\underline{u}(n)$, chosen to test each algorithm's ability to cope with varying degrees of excitation, consisted of a sum of equal-amplitude complex sinusoids whose frequencies changed suddenly every 800 samples (see Fig. 2). To model finite word-length effects, the real and imaginary parts of each filter coefficient were rolled-over when their amplitudes exceeded 9.5. All experiments were conducted with step-size $\mu = 0.1$.

Fig. 3 compares the performance of the standard LMS algorithm with and without numerical overflow. MSE drastically increased whenever unmitigated parameter drift induced numerical roll-over. Brief MSE spikes also occurred just after the input signal experienced a sudden change and before the adaptive filter had a chance to reconverge. In Fig. 4, we

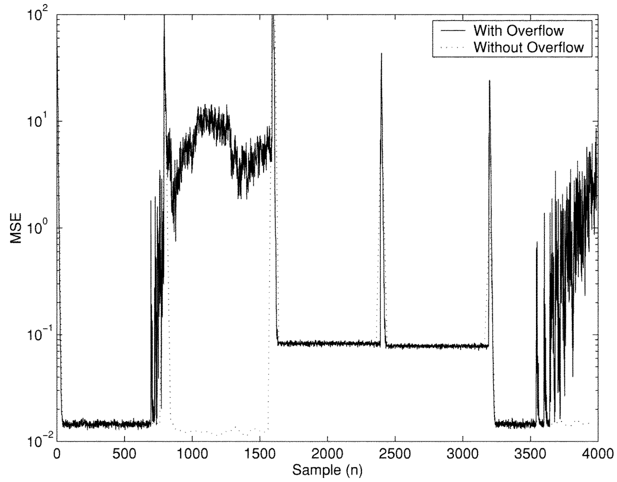

Fig. 3. Standard LMS with and without overflow effects.

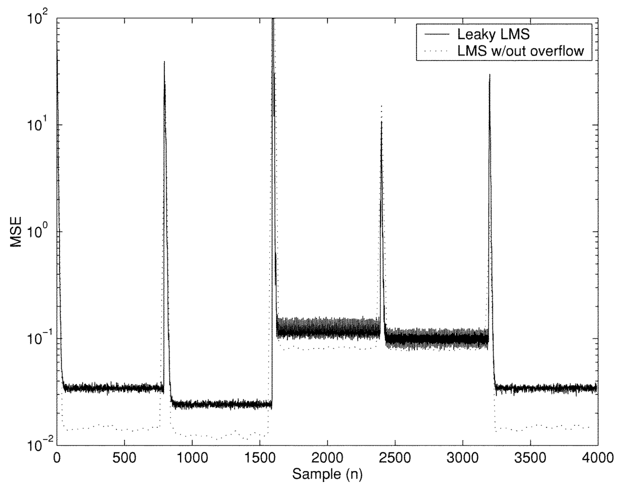

Fig. 4. MSE trajectory of L-LMS.

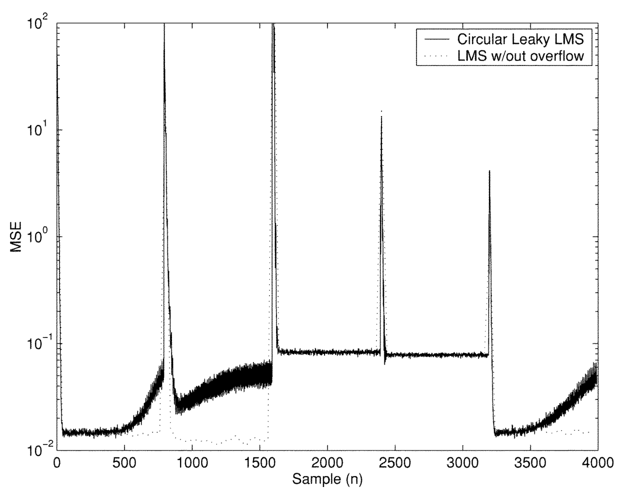

Fig. 5. MSE trajectory of CL-LMS.

implemented L-LMS with $\gamma = 0.02$. While L-LMS successfully avoids roll-over, the leakage-induced bias clearly increases MSE. Fig. 5. shows CL-LMS implemented using $\{\alpha_0, C_1, C_2, D\} = \{1, 7.5, 8.5, 0.5\}$. At first, CL-LMS introduces less bias than L-LMS. When the parameters drift near the

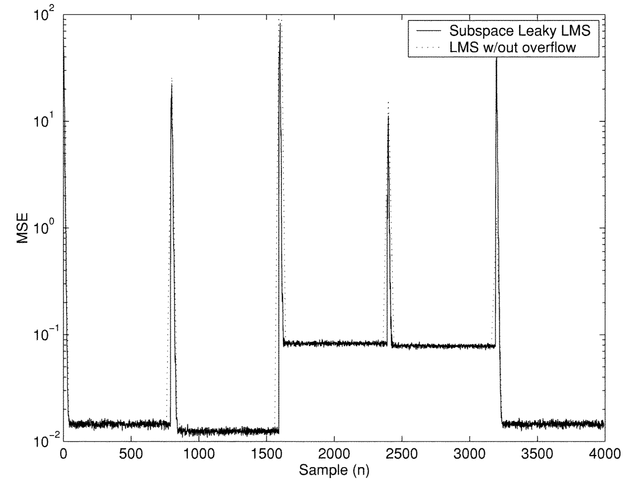

Fig. 6. MSE trajectory of SL-LMS.

bounds set by CL-LMS, however, leakage is frequently applied to prevent further drift and hence MSE increases. Finally, Fig. 6 shows SL-LMS applied to the same system using interleak interval $N = 60$ and DFT reinitialization set $\mathcal{S}$. Here we see that the SL-LMS algorithm outperforms the other algorithms by yielding essentially the same MSE behavior as standard LMS *without numerical overflow*.

V. CONCLUSION

This letter presented SL-LMS, a new adaptive filtering algorithm designed to mitigate the problem of parameter drift induced by coefficient update noise in the presence of nonpersistently exciting input signals. While tracking the unexcited subspace of the input signal, SL-LMS attempts to leak the adapted filter taps only within that subspace. This approach can yield a low-drift adaptive filtering system with low parameter bias. While the computational cost of SL-LMS is greater than that of CL-LMS or L-LMS, it is still linear in the filter parameters. Simulations have suggested that SL-LMS offers MSE performance superior to that of previously proposed leakage algorithms.

REFERENCES

[1] B. Widrow and M. E. Hoff, Jr., "Adaptive switching circuits," in *Proc. IRE WESCON Conf. Rec.*, 1960, pp. 96–104.
[2] B. Widrow, J. McCool, and M. Ball, "The complex LMS algorithm," *Proc. IEEE*, pt. 63, pp. 719–720, 1975.
[3] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 1996.
[4] W. A. Sethares *et al.*, "Parameter drift in LMS adaptive filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 868–879, 1986.
[5] P. A. Ioannou, "Robust adaptive control," in *Proc. 6th Yale Workshop Adaptive and Learning Systems*. New Haven, CT, 1975, pp. 719–720.
[6] K. Mayyas and T. Aboulnasr, "The leaky LMS algorithm: MSE analysis for Gaussian data," *IEEE Trans. Signal Processing*, vol. 45, pp. 927–934, Apr. 1997.
[7] V. H. Nascimento and A. H. Sayed, "Unbiased and stable leakage-based adaptive filters," *IEEE Trans. Signal Processing*, vol. 47, pp. 3261–3276, Dec. 1999.
[8] S. C. Douglas, "Performance comparison of two implementations of the leaky LMS adaptive filter," *IEEE Trans. Signal Processing*, vol. 45, pp. 2125–2129, Aug. 1997.
[9] N. R. Yousef and A. H. Sayed, "A unified approach to the steady-state and tracking analyzes of adaptive filters," *IEEE Trans. Signal Processing*, vol. 49, pp. 314–324, Feb. 2001.