Analysis of stochastic gradient tracking of time-varying polynomial Wiener systems

Author
Celka, Patrick

Published
2000

Journal Title
IEEE Transaction on Signal Processing

DOI
https://doi.org/10.1109/78.845925

Copyright Statement
Copyright 2000 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Downloaded from
http://hdl.handle.net/10072/65962
Analysis of Stochastic Gradient Tracking of Time-Varying Polynomial Wiener Systems

Neil J. Bershad, Fellow, IEEE, Patrick Celka, and Jean-Marc Vesin

Abstract—This paper presents analytical and Monte Carlo results for a stochastic gradient adaptive scheme that tracks a time-varying polynomial Wiener system [i.e., a linear time-invariant (LTI) filter with memory followed by a time-varying memoryless polynomial nonlinearity]. The adaptive scheme consists of two phases: 1) estimation of the LTI memory using the LMS algorithm and 2) tracking the time-varying polynomial-type nonlinearity using a second coupled gradient search for the polynomial coefficients. The time-varying polynomial nonlinearity causes a time-varying scaling for the optimum Wiener filter for Phase 1. These time variations are removed for Phase 2 using a novel coupling scheme to Phase 1. The analysis for Gaussian data includes recursions for the mean behavior of the LMS algorithm for estimating and tracking the optimum Wiener filter for Phase 1 for several different time-varying polynomial nonlinearities and recursions for the mean behavior of the stochastic gradient algorithm for Phase 2. The polynomial coefficients are shown to be accurately tracked. Monte Carlo simulations confirm the theoretical predictions and support the underlying statistical assumptions.

Index Terms—Adaptive estimation, adaptive filters, adaptive systems, identification, nonlinear filters, nonlinear systems, signal processing.

I. INTRODUCTION

MUCH research has been performed over the recent years on the nonlinear system identification problem. Many methods have been developed and devoted to this task. Orthogonal least square methods and fast associated algorithms have been developed for nonlinear system identification using NARMAX or orthogonal or nonorthogonal series expansions [1]–[5]. Neural networks (perceptron- and radial basis function-based) have been widely used for this problem and provide good results [6]–[8]. Some analytical convergence results for one- or two-layer perceptrons have been obtained when used for nonlinear channels with or without memory [9]–[13].

Recently, the stochastic gradient adaptive identification of Wiener systems (a linear filter followed by a zero-memory nonlinear function) with noisy input and output measurements has been analyzed [14]. Scalar input and output signals \( x(n) \) and \( y(n) \) were corrupted by zero-mean input and output Gaussian noises. Two identification schemes were studied and shown to perform satisfactorily when the input noise is not too large. The analytical results were obtained by modeling a smooth threshold type nonlinearity with an Erf function with input and output scaling factors. Many nonlinear systems can be modeled globally using this family of nonlinear functions. This includes communication channels, ionic currents through biological cell membranes, and electronic amplifiers. This work has been extended most recently to the gradient identification of Wiener–Hammerstein systems (a linear filter followed by a zero-memory nonlinear function followed by a second linear filter) [15].

The nonlinear portion of the Wiener system often operates in a small region about a bias point. It is then simpler to study the identification of Wiener systems using a linear filter followed by a limited Taylor series expansion around the bias point. The unknown system output is obscured by some noise \( n(n) \). The identification of polynomial-type nonlinear systems with memory and this structure can be handled with this model. Identification is performed in two steps: 1) linear filter identification using one LMS algorithm and 2) polynomial nonlinearity identification using a second LMS algorithm. LMS algorithms can be used to identify both the linear and the polynomial coefficients since the unknown coefficients of the polynomial nonlinearity are linearly imbedded in the model. The statistical behaviors of these schemes were studied in [16]. Recursions for the mean polynomial coefficients were obtained. An approximation for the polynomial coefficient fluctuation behavior was computed and yielded satisfactory agreement with Monte Carlo simulations.

This paper extends the approach used in [16] to the study of the tracking of time-varying polynomial systems with memory, initially again using the two gradient search algorithms described above. However, the time-varying polynomial nonlinearity causes a time-varying scaling for the optimum Wiener filter for the linear filter identification part. This implies that the LMS algorithm must be able to track these time variations. A mean tracking analysis is performed for small \( \mu \) (slow learning) for several time-varying polynomial models. Closed-form expressions are obtained for the mean steady-state weights of the LMS algorithm. Recursions are derived for the mean behavior of the polynomial coefficient estimator and tracker using these results. These recursions cannot be solved in closed form because of the fundamental nonlinear nature of the problem. However, the recursions are solved iteratively and shown to be in good agreement with the Monte Carlo
simulations. This supports the validity of the mathematical assumptions used to obtain these recursions. However, the recursions do not predict good tracking performance for this polynomial estimator. A second scheme is presented that orthogonalizes the observations and removes the time-varying Wiener filter scaling parameter prior to tracking the polynomial variations.

There were two steps in this research effort. The first step tried to extend the approach in [16] for the fixed polynomial case directly to the time-varying polynomial case. There were two adaptation phases in [16]. In the first phase, the linear portion of the channel \( H \) was estimated using the LMS algorithm. The mean of the LMS algorithm converged to the correct theoretical scale factor times \( H \) with a small variance. The weights were then frozen. Then, the polynomial coefficients were estimated using a second gradient adaptation. Two problems arose when this procedure was applied to the time-varying case. The first problem was the filtering effects of the first LMS procedure. This problem was analyzed and overcome, hence the material in Section II-B. However, this led to a time-varying scale factor on the estimate of the linear filter \( H \) (see Fig. 2). When some simulations were tried, this time-variation caused havoc with the second gradient search for the time-varying polynomial coefficients. This problem led to the material at the beginning of Section IV.

The second problem was that the second gradient search [as discussed above; see (48)] caused coupling between the components of \( a(n) \) (\( a(n) \) is the gradient estimate of the vector of the polynomial coefficients [see Fig. 1 and (1)]). The coupling was due to 1) the correlation in the observations in the polynomial model of the nonlinearity and 2) the time variations in \( a(n) \). Hence, the approach of [16] was discarded.

The next step was to try to come up with an effective scheme to overcome the coupling. The Hermite polynomial expansion in (48) orthogonalized the observations and eliminated the correlation. The time variations in \( H \) were removed by the scheme in Fig. 7 but requires parallel processing of the data.

This paper is organized in the following manner. Section II presents a review of some previous results for the fixed polynomial case [16] and some new analytic results for the linear filter mean tracking. Section III investigates the behavior of the nonlinear stochastic gradient learning algorithm and derives mean recursions for the polynomial coefficient estimator and tracker. Section IV shows that the tracking scheme presented in Section III is not effective and derives a new tracking scheme. Subsequent Monte Carlo simulations support the theory and demonstrate the effectiveness of the new scheme. Section V presents the conclusions.

II. SOME PRELIMINARIES

A. Linear Filter Estimation and Tracking

The unknown system structure consists of an \( N \)th-order linear time-invariant system \( H \) followed by a time-varying memoryless nonlinearity. The output of the linear system is \( z(n) = H^T X(n) \), where \( X(n) \) is the input [see (4)]. The nonlinearity is assumed to be represented by a \( P \)th-order \((P > 0)\) time-varying Taylor expansion \( f[x(n), a(n)] \) around a bias point (see Fig. 1). The \( a(n) \) are the coefficients of the polynomial expansion. This system is a Wiener-type block structure [17]. Consider the identification and tracking of a local expansion of the time-varying nonlinearity about a bias point. The input and output sequences \( x(n) \) and \( y(n) \) are composed of \( L \) samples with power \( \sigma_x^2 \) and \( \sigma_y^2 \) respectively. The output is corrupted by additive noise \( n_0(n) \) with power \( \sigma_0^2 \). The signals \( x(n) \) and \( n_0(n) \) are zero-mean white Gaussian processes, independent of each other. Hence

\[
y(n) = f[z(n), a(n)] + n_0(n)
\]

(1)

where

\[
f[x, a(n)] = \sum_{k=0}^{P} a_k(n)x^k
\]

(2)

and

\[
a^T(n) = [a_0(n), a_1(n), \ldots, a_P(n)]
\]

(3)

The output of the linear filter is \( z(n) = H^T X(n) \), where

\[
H^T = [H_1, H_2, \ldots, H_N]
\]

(4a)

and

\[
X^T(n) = [x(n), x(n - 1), \ldots, x(n - N + 1)]
\]

(4b)

Since \( x(n) \) is white, \( z(n) \) has power \( \sigma_z^2 = \sigma_x^2 H^T H \). The goal is to estimate the \( N \) fixed parameters of \( H \) and estimate and track the \( P + 1 \) time-varying parameters of \( a(n) \).

Let \( E[x] \) denote the expectation of \( x \). The optimum time-varying Wiener filter \( H_0(n) \) for this problem satisfies the orthogonality condition

\[
E[e_{H_0(n)}X(n)] = E \left[ \{y(n) - H_0^T(n)X(n)\} X(n) \right] = 0
\]

(5)

where

\[
e_{H_0(n)} = y(n) - H_0^T(n)X(n).
\]

The solution to (5) is time varying and given by

\[
H_0(n) = c(n)H
\]

(6)
where
\[
\alpha(n) = E \left[ \left( \frac{\partial f[x, \alpha(n)]}{\partial x} \right) \right] | x = z(n).
\] (7)
Equation (7) follows from Bussgang’s theorem [18]. Thus, the optimum filter is a time-varying scaled version of \( H \). Equation (7) has been evaluated in [16, eq. (11)] for zero-mean Gaussian data \( z(n) \)
\[
\alpha(n) = \sum_{k=0}^{[(P-1)/2]} (2k+1)(2k-1)!! \sigma_z^{2k} \alpha_{2k+1}(n) \tag{8}
\]
where \( n!! = 1 \cdot 3 \cdot 5 \cdots n \), and \([k]_T \) denotes the smallest integer nearest \( k \). Thus, the time-varying scale factor \( \alpha(n) \) is a function of the odd time-varying coefficients of the polynomial nonlinearity. Thus, even if \( H \) is time invariant, the optimum Wiener filter \( H_0(n) \) may be time varying, as can be seen from (6) and (8).

### B. Tracking Using the LMS Algorithm

The LMS weight vector \( W(n) \) is given by [19]–[22]
\[
W(n+1) = W(n) + \mu e(n)X(n)
\] (9)
where
\[
e(n) = y(n) - W^T(n)X(n).
\] (10)
A recent paper [14] has shown that \( H \) (time invariant) can be identified to within a scale factor using the LMS algorithm when the nonlinearity is time invariant (see Fig. 1). Furthermore, the LMS mean weight vector \( E[W(n)] \) converges to the optimum Wiener filter, i.e., \( \lim_{n \to \infty} E[W(n)] = H_0 \). Here, \( H_0(n) \) is time varying, and thus, this result is not applicable. One would not expect this equality in the time-varying case. The LMS adaptive filter must track the time variations in \( H_0(n) \). The time variations will cause the algorithm to suffer some tracking delay. This delay will manifest itself on average in \( E[W(n)] \).

Inserting (10) in (9) yields
\[
W(n+1) = \left[ I - \mu X(n)X^T(n) \right] W(n) + \mu e(n)X(n)
\]
\[
\times \left\{ \sum_{k=0}^{P} a_k(n) \left[ H^T X(n) \right]^k + n_0(n) \right\} X(n).
\] (11)
Averaging both sides of (11) and using the well-known LMS independence theory assumption [21] (i.e., the input data vector \( X(n) \) is an independent vector sequence) yields
\[
E[W(n+1)] = \left[ I - \mu R_{XX} \right] E[W(n)] + \mu \sum_{k=0}^{P} a_k(n)
\]
\[
E \left[ H^T X(n) \right]^k X(n)
\] (12)
where \( R_{XX} = E[X(n)X^T(n)] = \sigma_z^2 I \) for analytic simplicity. The expectations inside the sum are evaluated as follows. Since \( X(n) \) is a zero mean Gaussian vector, \( H^T X(n) \) is a Gaussian random variable. Hence, the entire sum is an expectation of the form \( E[f(x_1, x_2)] \), where \( x_1 \) and \( x_2 \) are jointly Gaussian. This expectation can be evaluated using, again, Bussgang’s theorem [18] as in (7)
\[
E[W(n+1)] = \left[ I - \mu R_{XX} \right] E[W(n)] + \mu \alpha(n) R_{XX} H.
\] (13)
Thus, (13) has the closed-form solution
\[
E[W(n)] = \left( 1 - \mu \sigma_z^2 \right)^n E[W(0)]
\]
\[
+ \mu \sigma_z^2 \sum_{p=0}^{n-1} \left( 1 - \mu \sigma_z^2 \right)^{n-1-p} \alpha(p)H.
\] (14)
Equation (14) clearly displays the mean filtering effects of the LMS algorithm on the time-varying input \( \alpha(n)H \). Let \( k(n) \) denote the time-varying mean scale factor on \( H \) to which the mean of the LMS algorithm approaches as \( n \) becomes large
\[
E[W(n)] = k(n)H.
\] (14a)
In general, \( k(n) \neq \alpha(n) \) because of the filtering effect of LMS. It is now necessary to specify the time variations of \( \alpha(n) \) in order to proceed further.

1) Tracking Sinusoidal Variations in \( \alpha(n) \): Let the variations in \( \alpha(n) \) be sinusoidal
\[
a_4(n) = A_4 \cos(\omega_4 n + \phi_4).
\] (15)
Assuming (14) converges, inserting (15) in (14), and letting \( n \) grow without bound yields the time-varying steady-state solution
\[
E[W(n)] = \left( \frac{\mu \sigma_z^2}{2} \right)^{[P-1]/2} \sum_{k=0}^{P-1} (2k+1)(2k-1)!! \sigma_z^{2k} A_{2k+1}.
\]
\[
\times \left[ Z(j \omega_{2k+1}) \exp \left\{ j(\omega_{2k+1} n + \phi_{2k+1}) \right\} \right.
\]
\[
+ \left. Z(-j \omega_{2k+1}) \exp \left\{ -j(\omega_{2k+1} n + \phi_{2k+1}) \right\} \right] H
\]
\[
=k(n)H.
\] (16)
where
\[
Z(j \omega) = \frac{1}{[\mu \sigma_z^2 \left( 1 - \frac{\sigma_z^2}{\sigma_z^2} \right)]}.
\] (17)
\( Z(j \omega) \) represents the attenuation and phase shift introduced by the filtering behavior of the mean LMS algorithm weights in (14). The tracking delay is displayed in the phase shift. In general, \( k(n) \neq \alpha(n) \). In principle, if the \( \omega_k \) were known \( a \) priori, then the individual \( A_k \) could be estimated by appropriately filtering \( W(n) \).

2) Tracking Linear Time Variations in \( \alpha(n) \): Suppose
\[
f[x, \alpha(n)] = n \sum_{k=0}^{P} a_k x^k.
\] (18)
Then
\[
\alpha(n) = n \sum_{k=0}^{[(P-1)/2]} (2k+1)(2k-1)!! \sigma_z^{2k} a_{2k+1} = n \alpha_0
\] (19)
where \( \alpha_0 \) is the value of the summation in (19). Inserting (19) in (14) yields
\[
E[W(n)] = \left( 1 - \mu \sigma_z^2 \right)^n E[W(0)]
\]
\[
+ \mu \sigma_z^2 \alpha_0 \sum_{p=0}^{n-1} \left( 1 - \mu \sigma_z^2 \right)^{n-1-p} pH.
\] (20)
\( 1 \) This will occur, for example, if \( \alpha(n) \) is bounded, and \( 1 - \mu \sigma_z^2 < 1. \)
\( 2 \) \( E[W(n)] \) can be viewed as the response of a first-order system with time constant \( 1 - \mu \sigma_z^2 \) to periodic inputs. If the frequencies of the input are known, then \( E[W(n)] \) can be filtered to separate out the different frequencies. The amplitude of each of the filtered sinusoids can be used to estimate the \( A_k \)’s.
Assuming (20) converges, summing on \( p \), and letting \( n \) become large yields

\[
E[W(n)] = \left(n - \frac{1}{\mu \sigma_x^2}\right) \alpha_0 H. \tag{21}
\]

Equation (21) displays a tracking delay inversely proportional to \( \mu \), as expected.\(^3\)

3) Tracking Time Variations in the Even Coefficients: Suppose that only the even coefficients of \( a(n) \) are time varying, i.e.,

\[
f[x, a(n)] = \sum_{k \text{ even}}^P a_k(n)x^k + \sum_{k \text{ odd}}^P a_k(n)x^k. \tag{22}
\]

Then

\[
\alpha(n) = \sum_{k=0}^{(P-1)/2} (2k+1)(2k-1)! \sigma_x^{2k} a_{2k+1} = \alpha_0 \tag{23}
\]

is time invariant. Hence, (14) becomes

\[
E[W(n)] = (1 - \mu \sigma_x^2)E[W(0)]
+ \mu \sigma_x^2 \alpha_0 \sum_{j=0}^{n-1} (1 - \mu \sigma_x^2)^{n-1-j} H. \tag{24}
\]

Assuming (24) converges, summing on \( p \), and letting \( n \) approach infinity yields

\[
\lim_{n \to \infty} E[W(n)] = \alpha_0 H. \tag{25}
\]

Hence, for this time-varying case, the LMS mean weight converges to the optimum Wiener filter.

4) Effect of Weight Fluctuations: The previous subsections studied the behavior of the LMS mean weights for various time variations in \( f[x, a(n)] \). The LMS weights display fluctuations about these mean values because the input is stochastic. This leads to some problems in the analysis of the nonlinearity learning. For instance, the output \( \hat{y}(n) = W(n)x(n) \). This is somewhat different from \( k(n)^T X(n) \). Nevertheless, under the slow convergence hypothesis (small \( \mu \)), the output \( \hat{z}(n) \) can be written approximately as \cite{14}

\[
\hat{z}(n) \approx k(n)^T X(n). \tag{26}
\]

C. Scaling Property for the Nonlinearity

The parameter vector \( a(n) \) is unknown. Thus, \( k(n) \) is also unknown. It is not possible to identify \( a(n) \) and \( k(n) \) independently \cite{16}. Indeed, denoting \( Z^T = (1, z, z^2, \ldots, z^P) \), (2) can be rewritten as

\[
f[z, a(n)] = a^T(n) Z. \tag{27}
\]

\( f[z, a(n)] \) is identified using a similar polynomial-type function \( f[\hat{z}, \hat{a}(n)] \), where \( \hat{z}(n) = \hat{H}^T(n)X(n) \approx k(n)H^T X(n) \) (see Fig. 2) and is given by

\[
f[\hat{z}, \hat{a}(n)] = \sum_{k=0}^P \hat{a}_k(n)\hat{z}^k = \hat{a}^T(n)\hat{Z}. \tag{28}
\]

with \( \hat{a}^T(n) = (\hat{a}_0, \hat{a}_1, \hat{a}_2, \ldots, \hat{a}_P) \). Introduce the diagonal matrix \( G(n) \)

\[
G(n) = \text{diag} \left[1, k(n), k^2(n), k^3(n), \ldots, k^P(n)\right]. \tag{29}
\]

Using (26), (27), and (29), (28) can rewritten as

\[
f[\hat{z}(n), \hat{a}(n)] = \hat{a}^T(n)\hat{Z}(n)
= \hat{a}^T(n)G(n)\hat{Z}(n)
= f[z(n), G(n)\hat{a}(n)]. \tag{30}
\]

Perfect identification occurs when

\[
f[\hat{z}(n), \hat{a}(n)] = f[z(n), G(n)\hat{a}(n)] = f[z(n), a(n)]. \tag{31}
\]

Thus

\[
\hat{a}(n) \approx G^{-1}(n)a(n). \tag{32}
\]

Equation (32) implies that only \( \hat{a}^T(n) = (a_0(n), a_1(n)/k(n), a_2(n)/k^2(n), \ldots, a_P(n)/k^P(n)) \) can be estimated \cite{23}, [24]. Thus, it is impossible to identify \( k(n) \) and \( a(n) \) independently. Equation (32) is approximate due to the approximations in (26) and (30).

III. LEARNING THE NONLINEARITY

A. Global Stationary Points

The \( N \) coefficients of the linear part of the model \( H \) have been estimated in Section II using the LMS algorithm. Now, the \( P+1 \) parameters of \( \hat{a}(n) \) are estimated using a second stochastic gradient learning algorithm. The estimate of \( a(n) \), \( \hat{a}(n) \), converges in some statistical sense toward the minimum of the mean square error surface \cite{19}

\[
\xi_F = E \left[ e^2 (n) \right] \equiv \xi_F[a(n)]. \tag{33}
\]

where

\[
e_F(n) = y(n) - f (\hat{z}(n), \hat{a}(n)). \tag{34}
\]

Using (1), (27), and (30), (33) can be written as

\[
e_F(n) = f(z(n), a(n)) - f (\hat{z}(n), \hat{a}(n)) + n_0(n)
= a^T(n)Z(n) - \hat{a}^T(n)\hat{Z}(n) + n_0(n)
= (a(n) - G(n)\hat{a}(n))^T Z(n) + n_0(n). \tag{35}
\]

where \( \hat{Z}(n) \) is approximated by \( G(n)Z(n) \). The parameter vector \( a(n) \) is time dependent. Equation (33) defines a \( P+1 \)-dimensional surface. This surface is unimodal and possesses a time-varying global minimum at \( a^*(n) \). The stochastic gradient algorithm will attempt to track this minimum for step sizes inside some convergence region. Define the vector

\[
V_F(n) = \hat{a}(n) - G^{-1}(n)a(n). \tag{36}
\]

\( V_F \) is denoted the parameter deviation vector (PDV). Using (36), (35) can be rewritten as

\[
e_F(n) \approx -V_F^T(n)\hat{Z}(n) + n_0(n). \tag{37}
\]

Equation (33) for the MSE surface becomes

\[
\xi_x = \sigma^2 + E \left[ V_F^T(n)\hat{Z}(n)V_F(n) \right]
- 2E \left[ V_F^T(n)\hat{Z}(n)n_0(n) \right]. \tag{38}
\]
Assuming that \( n_{o}(n) \) and \( \hat{Z}(n) \) are independent (see [20, p. 61], for example, and Section III)

\[
Q_\epsilon = \sigma_\epsilon^2 + V_f^T(n)E[\hat{Z}(n)\hat{Z}(n)^T]V_f(n)
= \sigma_\epsilon^2 + V_f^T(n)E[Q(n)]V_f(n),
\]

(39)

\( Q(n) = \hat{Z}(n)\hat{Z}(n)^T \equiv G(n)Z(n)Z(n)^T \) is a square semi-definite positive symmetric real matrix. Equation (39) defines a \( P + 1 \) quadratic error surface. This surface is elliptic and possesses a unique global minimum at \( V_f = 0 \) or, using (36), at \( a^*(n) = G(n)a(n) \). The stochastic gradient algorithm will track this minimum in some statistical sense under some stability conditions. \( E[Q(n)] \) involves expectation values of the form \( E[\hat{s}_i(n)\hat{s}_j(n)] \) for \( i, j = 0, \cdots, P \). The matrix \( E[Q(n)] \) plays the role of the input vector covariance matrix in standard linear LMS theory [19], [20]. The matrix \( E[Q(n)] \) is easily computed for Gaussian inputs

\[
E[Q(n)] = G(n)
\begin{bmatrix}
1 & 0 & \sigma_\epsilon^2 & 0 \\
0 & \sigma_\epsilon^2 & 0 & 3\sigma_\epsilon^4 \\
3\sigma_\epsilon^4 & 0 & (2P - 1)!!\sigma_\epsilon^{2P}
\end{bmatrix}G(n)
= G(n)RG(n),
\]

(40)

where \( \sigma_\epsilon^2 \) is given in (3). The last row and column of \( E[Q(n)] \) depends on the parity of \( P \).

B. Stochastic Gradient Algorithm

A stochastic gradient algorithm was used in [16] (see Fig. 2) to estimate a fixed vector after estimating \( H \), as in Fig. 1. This scheme is proposed here as well. The adaptive gradient recursion for \( \hat{a}(n) \) is

\[
\hat{a}(n+1) = \hat{a}(n) + M_e(n)\hat{Z}(n)
= \left[ I - M\hat{Z}(n)\hat{Z}(n)^T \right] \hat{a}(n) + Mn_o(n)\hat{Z}(n)
+ M\hat{Z}(n)\hat{Z}(n)^T \hat{a}(n),
\]

(41)

where \( M = \text{diag}[\mu_0, \mu_1, \mu_2, \cdots] \), \( \mu_i > 0 \) is the algorithm step size matrix [16, eq. (37)]. Not surprisingly, the recursions for \( \hat{a}(n) \) are linear because \( f \) is linear in \( a(n) \) (or, equivalently, \( \hat{X}_f \) is quadratic in \( V_f \)). Equation (41) is used to learn \( a(n) \) after the filter coefficients for \( H \) have been computed, as described in Section II.

C. Mean Behavior of (41)

Averaging (41) and using the independence assumption (i.e., inputs at different times are statistically independent [18], [20], [25]–[27]) yields

\[
E[\hat{a}(n+1)] = \left[ I - ME\left[\hat{Z}(n)\hat{Z}(n)^T\right] \right] E[\hat{a}(n)]
+ ME\left[\hat{Z}(n)\hat{Z}(n)^T\right] a(n)
= \left[ I - MG(n)E\left[\hat{Z}(n)\hat{Z}(n)^T\right] G(n) \right] E[\hat{a}(n)]
+ MG(n)E\left[\hat{Z}(n)\hat{Z}(n)^T\right] a(n)
= \left[ I - MG(n)RG(n) \right] E[\hat{a}(0)] + MG(n)Ra(n),
\]

(42)

In general, (42) represents a linear time-varying matrix recursion with a time-varying input. It is difficult to find analytic solutions to such recursions. Some special cases for \( G(n) \) were considered in Section II and will now be studied here.

1) Time Variations in the Even Coefficients: \( G(n) \) is time invariant as can be seen from (23) and (29). Then, (42) simplifies to

\[
E[\hat{a}(n+1)] = \left[ I - MG\hat{G} \right] E[\hat{a}(n)] + MG\hat{R}a(n).
\]

Equation (43) is a linear time-invariant matrix recursion for \( E[\hat{a}(n)] \) with closed-form solution

\[
E[\hat{a}(n)] = \left[ I - MG\hat{G} \right]^n E[\hat{a}(0)]
+ \sum_{m=0}^{n-1} \left[ I - MG\hat{G} \right]^{n-m-1} MG\hat{R}a(m).
\]

(44)

\( MG\hat{G} \) is a square symmetric semi-definite positive matrix. \( \hat{G} \) can be written in terms of its eigenvalues and eigenvectors as \( G\hat{R}_\Delta = S\Delta S^{-1} \), where \( \Delta = \text{diag}[^\lambda_1, ^\lambda_2, \cdots, ^\lambda_N] \) is the diagonal matrix of eigenvalues, and \( S \) is the matrix of orthogonal eigenvectors. Thus, (44) can be rewritten as

\[
E[\hat{a}(n)] = \left[ I - MS\Delta S^{-1} \right]^n E[\hat{a}(0)]
+ \sum_{m=0}^{n-1} \left[ I - MS\Delta S^{-1} \right]^{n-m-1} MS\Delta S^{-1} G^{-1} a(m).
\]

(45)

This is as far as we can proceed analytically without finding \( \Delta \) and \( S \) and specifying \( a(n) \).

2) Linear Time Variations in \( a(n) \): Using (21) for large \( n \)

\[
G(n) = \left( n - \frac{1}{\mu_\Delta^2} \right) G_0
\]

(46)

where \( G_0 \) is defined by (29) with \( \alpha(n) \) replaced by \( \alpha_0 \). Inserting (19) and (46) in (42) yields

\[
E[\hat{a}(n+1)] = \left[ I - \left( n - \frac{1}{\mu_\Delta^2} \right)^2 MG_0\hat{R}G_0 \right] E[\hat{a}(n)]
+ n \left( n - \frac{1}{\mu_\Delta^2} \right) MG_0\hat{R}a(0).
\]

(47)

Equation (47) is a linear time-varying matrix recursion for \( E[\hat{a}(n)] \) with no closed-form solution. In addition, the question of stability of the recursion arises for large \( n \).

3) Sinusoidal Variations in \( a(n) \): This case is the most difficult to study because each component of \( G(n) \) undergoes different time variations [see (16) and (29)]. Thus, (42) cannot be simplified. Equation (42) is a linear time-varying matrix recursion for \( E[\hat{a}(n)] \) with no closed-form solution.

4) Solutions: Not too surprisingly, none of the above cases yield closed-form expressions for the mean weights. Thus, (42), (43), and (47) must be solved numerically. These deterministic solutions are displayed in the next section and compared with Monte Carlo simulation results.

IV. MODIFICATIONS TO THE THEORY IN SECTION III

The theory in Section III was developed as a natural extension of the work in [16] for the fixed polynomial nonlinearity. An initial set of ten Monte Carlo simulations was run to check the va-
Fig. 3. Monte Carlo simulation of (9) (thin lines) and theoretical prediction using (16) and (17) (bold lines).

Fig. 4. Monte Carlo simulation of (41) (thin lines) and theoretical prediction using (45) (bold lines).

Fig. 5. Monte Carlo simulation of (9) (thin lines) and theoretical prediction using (16) and (17) (bold lines).

Fig. 6. Monte Carlo simulation of (41) (thin lines) and theoretical prediction using (45) (bold lines).

lidity of this theory for the time-varying case. Ten Monte Carlo runs were run for each simulation result. In all cases, $H = [0.0201, 0.2301, 0.4981, 0.2301, 0.0201]$. The SNR = 10 dB, $\sigma_w^2 = 4.0$, and $f_0 = 100$. The adaptation was done in parallel for the time-varying case. The LMS algorithm for $H$ and the stochastic gradient algorithm for the polynomial coefficients were run simultaneously.

**Case 1—Sinusoidal Time Variations in the Even Coefficients:** The parameters selected were

$$a(n)^T = [0, 1, 0.5 \times \cos(2\pi f_0 n/f_s), 0.1]$$

where $f_0 = 0.05$

$$\mu_{\text{LMS}} = 0.0005$$

$$M = \text{diag}[0.01, 0.005, 0.0005, 0.00005]$$

Using these values and (8), $a_0 = a_1 + 3a_3a_2^3 = 1.4257$. Fig. 3 shows very good agreement between theory and simulations for both the LMS weight vector $W(n)$ and the polynomial estimate of $a(n), \hat{a}(n)$. The LMS weights quickly converge to $a_0$, as predicted by (25). Fig. 4 shows the evolution of the polynomial coefficients $\hat{a}(n)$. The even coefficients $\hat{a}_e(n)$ and $\hat{a}_o(n)$ display a large sinusoidal oscillation at the same frequency as $a_2(n)$. This is due to component coupling effects, as can be seen from (44). The matrix $R$ is not diagonal, as given in (40). $E[\hat{a}_e(n)]$ and $E[\hat{a}_o(n)]$ converge to $a_1/a_0 = 0.7013$ and $a_3/a_0^3 = 0.00345$, respectively.

**Case 2—Sinusoidal Time Variations in the Odd Coefficients:** The parameters selected were

$$a(n)^T = [0, 1, 0.5 \times \cos(2\pi f_0 n/f_s), 0.1]$$

$$\mu_{\text{LMS}} = 0.0005$$

$$M = \text{diag}[0.01, 0.005, 0.0005, 0.00005]$$

Fig. 5 shows very good agreement between theory and simulations for both the LMS weight vector $W(n)$ and the polynomial estimate of $a(n), \hat{a}(n)$. The LMS weight amplitudes are scaled as in Fig. 3 but oscillate at the same frequency as $a_2(n)$. Fig. 6 shows that MC runs of (41) and theoretical predictions of (42) do not match. Moreover, (41) does not reflect the behavior of $a(n)$ in any sense. Since $a_3(n)$ is time varying, the matrix $G(n)$ is also time varying and generates oscillations in all the polynomial coefficients. Moreover, odd and even coefficients of $\hat{a}(n)$ are coupled via $R$. 
Based on these two cases, it appears that (41) is not a very good estimator for $\alpha(n)$. As already noted, the simulations and (42) and (44) suggest that there is coupling between the driver term $\alpha(n)$ and the mean of the estimate $E[\hat{\alpha}(n)]$. Time variations in the components of $\alpha(n)$ couple into different components of $E[\hat{\alpha}(n)]$ because of the nondiagonal structure of $MG(n)RG(n)$. Furthermore, time variations of $G$ in (44) appear to create distortions in $E[\hat{\alpha}(n)]$. These experimental and theoretical problems suggest two modifications to the estimation procedure.

The first modification involves the observation vector. The $R$ matrix in (40) is nondiagonal. Consider the following orthogonalization procedure [28] to diagonalize $R$. Using (2)

$$f[z, a(n)] = \sum_{k=0}^{P} a_k(n) z^k = \sum_{k=0}^{P} b_k(n) H c_k(z) = f_1[z, b(n)]$$

(48)

where $H c_k$ is a Hermite polynomial of order $k$. The $H c_k$ are orthogonal polynomials with respect to the Gaussian weighting function [29]. For example, for $P = 3$

$$f_1[z, b] = b_0 + b_1 z + b_2 [z^2 - \sigma_z^2] + b_3 [z^3 - 3\sigma_z^2 z].$$

Thus, with the data vector

$$X_1^T = [1, z, z^2 - \sigma_z^2, z^3 - 3\sigma_z^2 z]$$

$$E[X_1 X_1^T] = \text{diag} \left[ \gamma_0, \sigma_z^2, 2\sigma_z^2, 6\sigma_z^2 \right]$$

$$= \text{diag} \left[ \gamma_0, \gamma_1, \gamma_2, \gamma_3 \right].$$

(51)

$a$ and $b$ are related through the triangular matrix transformation

$$b = C a = \begin{bmatrix} 1 & 0 & \sigma_z^2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} a.$$  

(52)

Fig. 7. Nonlinear adaptive identification for a dynamic nonlinearity-learning the linear filter in parallel (lower system) and learning the dynamic nonlinearity (upper system) with the linear filter slaved to the lower system.

Note that this diagonalization procedure is not the same as the orthogonal matrix procedure below (44). $C$ is not an orthogonal matrix, and the diagonal entries in (51) are not the eigenvalues of $R$. This procedure can easily be generalized for any $P$.

The second modification involves the scaling matrix $G$. There is no problem if $G$ is time invariant, as in [16] or in Section II-B3 [see (14)]. However, it is shown below that the present scheme is not effective when $G(n)$ is time varying. The gradient search procedure for estimating $\hat{\alpha}(n)$ is

$$\hat{\alpha}(n + 1) = \hat{\alpha}(n) + M c_f(n) \hat{X}_1(n)$$

$$+ M n_0(n) \hat{X}_1(n) + M \hat{X}_1(n) X_1^T(n) b(n).$$

(53)

The mean behavior of (53) is

$$E[\hat{\alpha}(n + 1)] = [I - MGTG] E[\hat{\alpha}(n)] + MGT \bar{b}(n)$$

where $\Gamma = \text{diag} \left[ \gamma_0, \gamma_1, \gamma_2, \cdots, \gamma_P \right]$.  

(54)

Because of the diagonal form of (54), the solution can be written explicitly as

$$E[\hat{\alpha}(n)] = \left[1 - \mu k^{2\gamma_i} \bar{n}_i \right]^n E[\bar{b}(0)] + \sum_{m=0}^{n-1} \mu k^{2\gamma_i} \bar{n}_i - 1 \mu k^{2\gamma_i} \bar{b}(m).$$

(55)

Note that $E[\hat{\alpha}(n)]$ depends only on $\bar{b}(n)$. If $\bar{b}(n) \equiv \bar{b}$, then

$$\lim_{n \to \infty} E[\hat{\alpha}(n)] = \bar{b}/k.$$  

(56)

If $\bar{b}$ and/or $\alpha_i$ is time varying, the sum in (55) is not easy to evaluate. Furthermore, it is not clear that $E[\hat{\alpha}(n)]$ will resemble $\bar{u}(n)$.

Hence, consider the modified estimation scheme shown in Fig. 7. The lower arm yields an estimate of $H$ whose mean value is described by (14). $E[W(n)] = \tau(n) H$. $\tau(n)$ is estimated dynamically from $W(n)$ as

$$\hat{\tau}(n) = + \sqrt{W(n)TW(n),}$$

(57)
The filter in the upper arm is then
\[
\frac{W(n)}{\sqrt{W(n)W(n)}} \approx \frac{H}{\sqrt{H^TH}} = \text{constant}, \quad (58)
\]

Hence, now, \( G(n) \) is a constant. However, it is not possible to estimate both \( H \) and \( a(n) \) uniquely just from the input and output observations. Scalings can be assigned to \( H \) or \( a(n) \) that affect \( G(n) \). This ambiguity was removed by assuming that \( a_1 = 1 \) in [16]. Here, it will be assumed that \( H^TH \) is known \textit{a priori}. Hence, without loss, set \( H^TH = 1 \). Hence, \( G(n) = I \). In essence, \( \hat{G}(n) \) is an estimate of \( k(n) \). Fig. 7 shows the output of the estimated linear filter as
\[
\hat{z}(n) = \frac{\hat{x}(n)}{\hat{p}(n)} = \frac{k(n)z(n)}{\hat{p}(n)} \approx z(n), \quad (58a)
\]

Under this condition [replacing \( \hat{b}(n) \) by \( \hat{K}(n) \) in (53) and proceeding as in (55)], (55) simplifies to
\[
E\left[ \hat{b}_i(n) \right] = [1 - \mu \gamma_i]^{n} E[\hat{b}_i(0)] \\
+ \mu \gamma_i \sum_{m=0}^{n-1} [1 - \mu \gamma_i]^{n-m-1} b_i(m). \quad (59)
\]

Equation (59) can be solved for sinusoidal \( b_i(n) \) as in (16) with (17) replaced by
\[
Z_i(j\omega) = \frac{1}{e^{j\omega} - (1 - \mu \gamma_i)I}. \quad (60)
\]

Equations (59) and (60) indicate that the gradient search behaves like a bank of first-order filters with different time constants for each sinusoidal component of \( b_i(n) \). If \( \mu \gamma_i \) is sufficiently small or \( \omega \) sufficiently large, the sinusoidal components will be attenuated differently during the estimation process. This attenuation factor will need to be corrected before generating an estimate of \( a(n) \). This will require knowledge of \( \omega \). This attenuation factor will also increase the MSE. It is the price the gradient search pays for tracking the sinusoidal variation.

Further simplification occurs if \( \mu \gamma_i \) is selected such that \( \mu \gamma_i = \mu \) for all \( i \). Finally, the estimate of \( a(n) \), say, \( \hat{a}(n) \), obtained from \( \hat{K}(n) \) for the four-dimensional case is given by
\[
\hat{a}(n) = C^{-1} \hat{b}(n) = \begin{bmatrix} 1 & 0 & -\gamma_1^2 & 0 \\
0 & 1 & 0 & -3\gamma_1^2 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \end{bmatrix} \hat{b}(n). \quad (61)
\]

V. NUMERICAL STUDIES

This section demonstrates that the LMS tracking recursions derived in Section II and the mean behavior tracking recursions for the nonlinear parameters derived in Section IV yield accurate results and good tracking of \( a(n) \). Monte Carlo simulations for the linear part for sinusoidal time variations in the even and odd coefficients were presented in the previous section. The theoretical mean behaviors predicted by (13) were run for these two cases discussed in Section II-B. Monte Carlo simulations of (11) and (53) were run for comparison purposes. Ten Monte Carlo simulations were also run for these simulations.

Case 1—Sinusoidal Time Variations in the Even Coefficients:
\[
\alpha(n)^T = [0, 1, 0.5 \times \cos(2\pi f_{0}/f_s), 0.1] \\
\mu_{LMS} = 0.0005 \\
M = \mu_{LMS}^{-1} \quad \text{with } \mu = 0.001.
\]

Monte Carlo simulations are now presented here for the modified nonlinear polynomial coefficient tracking scheme given in Section IV. Computer solutions for (54), (55), and (58) [with \( \hat{K}(n) \) replacing \( \hat{b}(n) \)] were run for the three cases discussed in Section II-B. Monte Carlo simulations of (11) and (53) were run for comparison purposes. The figures show the first three components of \( \hat{H} \) and all components of \( a(n) \), theory, and simulations. Ten Monte Carlo simulations were also run for these simulations.

Fig. 8 shows \( \hat{b}(n) \) and \( E[\hat{b}_i(n)] \) obtained using (53) and (59). The theory and Monte Carlo simulations show very good agreement. Fig. 9 shows \( \hat{a}(n) \) and \( \hat{a}(n) \). \( E[\hat{a}(n)] \) from (61). Note that since \( G(n) = I \), the estimated vector \( \hat{a}(n) \) should converge to \( a(n) \). This is the case for the even coefficients, \( \hat{a}_0(n) \) converges to 0 without coupling to \( \hat{a}_2(n) \). Monte Carlo runs for \( \hat{a}_0(n) \) match \( E[\hat{a}_0(n)] \) and match \( a(n) \) after including the amplitude and phase shift factors given in (60). These good tracking results should be compared with those in Fig. 6.

Case 2—Sinusoidal Time Variations in the Odd Coefficients:
\[
\alpha(n)^T = [0, 1, 0.5, 0.1 \times \cos(2\pi f_{0}/f_s), 0.1] \\
\mu_{LMS} = 0.0005 \\
M = \mu_{LMS}^{-1} \quad \text{with } \mu = 0.001.
\]

Fig. 8 shows \( \hat{b}(n) \) and \( E[\hat{b}_i(n)] \) obtained using (53) and (59). The theory and Monte Carlo simulations show very good agreement. Fig. 9 shows \( \hat{a}(n) \) and \( \hat{a}(n) \). \( E[\hat{a}(n)] \) from (61). Note that since \( G(n) = I \), the estimated vector \( \hat{a}(n) \) should converge to \( a(n) \). This is the case for the even coefficients, \( \hat{a}_0(n) \) converges to 0 without coupling to \( \hat{a}_2(n) \). Monte Carlo runs for \( \hat{a}_0(n) \) match \( E[\hat{a}_0(n)] \) and match \( a(n) \) after including the amplitude and phase shift factors given in (60). These good tracking results should be compared with those in Fig. 6.
Fig. 10 shows $\tilde{b}(n)$ and $E[\tilde{b}(n)]$ obtained using (53) and (59). Again, the theory and Monte Carlo simulations show very good agreement. Fig. 11 shows $a(n)$ and $\hat{a}(n)$. $E[\hat{a}(n)]$ from (61). Again, some amplitude and phase shift factors are shown for $\hat{a}_3(n)$. Fig. 12 shows $\hat{a}(n)$ for $\mu$ increased to 0.01. Close agreement between $a_3(n)$ and $\hat{a}_3(n)$ is seen. The transient is shorter, and the coefficient fluctuations are larger, but the filtering effect of (60) is nearly negligible.

Case 3—Linear Time Variations in All Coefficients:

\[
\begin{align*}
\alpha_k(n) &= \alpha_k + \alpha_k^{\text{max}} n / L
\end{align*}
\]

\[
\begin{align*}
\alpha_0 &= 0; & \alpha_0^{\text{max}} &= 0 \\
\alpha_1 &= 1; & \alpha_1^{\text{max}} &= 0.3 \\
\alpha_2 &= 0.5; & \alpha_2^{\text{max}} &= 0.5 \\
\alpha_3 &= 0.1; & \alpha_3^{\text{max}} &= 0.1 \\
\mu_{\text{LMS}} &= 0.0005 \\
M &= \mu L^{-1} & \text{with} & \mu = 0.001.
\end{align*}
\]
VI. CONCLUSIONS

This paper has investigated the identification and tracking of a Wiener-type time-varying nonlinear system for Gaussian inputs using adaptive stochastic gradient algorithms. The estimation technique presented in [16] was not extendable to the time-varying case. A new scheme was developed. The linear and nonlinear polynomial parameters had to be identified separately using two coupled stochastic gradient search procedures. Although the unknown time-varying nonlinearity was locally expandable in a Taylor series, the nonlinear portion of the adaptive system had to be expanded in a set of Hermite polynomials to eliminate time-vary coupling effects. This representation orthogonalized the input to the nonlinear portion of the adaptive system for Gaussian data.

Analysis of the mean behavior of the two coupled stochastic gradient algorithms has been performed for a variety of time-varying polynomials. Monte Carlo simulations have confirmed the theoretical predictions for the linear and nonlinear portions of the unknown Wiener model. It has been demonstrated that the proposed new scheme can effectively track sinusoidal and linear polynomial time variations.

REFERENCES


Neil J. Bershad (S’60–M’62–SM’81–F’88) received the B.E.E. degree from the Rensselaer Polytechnic Institute (RPI), Troy, NY, in 1958, the M.S. degree in electrical engineering from the University of Southern California, Los Angeles, in 1960, and the Ph.D. degree in electrical engineering from RPI in 1962. He joined the Faculty of the School of Engineering, University of California, Irvine, in 1966 and is now an Emeritus Professor of Electrical and Computer Engineering. He was a Visiting Professor of electrical engineering at ENSEEIH-GAPSE, Toulouse, France, from 1994 to 1998, at the Signal Processing Laboratory, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, from 1997 to 1999; and at the University of Edinburgh, Edinburgh, U.K., in 1999. His research interests have involved stochastic systems modeling and analysis. His recent interests are in the area of stochastic analysis of adaptive filters. He has published a significant number of papers on the analysis of the stochastic behavior of various configurations of the LMS adaptive filter. His present research interests include the statistical learning behavior of adaptive filter structures for nonlinear signal processing, neural networks when viewed as nonlinear adaptive filters, and active acoustic noise cancellation.

Dr. Bershad has served as an Associate Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS in the area of phase-locked loops and synchronization. He was also an Associate Editor of the IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING in the area of adaptive filtering. He is presently a Member of the Signal Processing Theory and Methods Committee of the IEEE Signal Processing Society.

Patrick Celka received the M.Sc. degree in physical sciences from the Catholic University of Louvain-la-Neuve, Louvain-la-Neuve, Belgium, in 1987. He received the master degree in information and signal processing from the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, in 1993 and the Ph.D. degree in 1995 from EPFL under the Circuits and Systems Chair of Prof. M. Hasler.

He has worked as a Senior Research Assistant at the Signal Processing Laboratory under Prof. M. Kunt at EPFL. He was involved in research in neuro-cardiovascular systems and signal modeling and analysis. He is currently leading a biomedical project dealing with the automatic detection/classification of EEG seizures in newborns at the Signal Processing Research Centre, Queensland University of Technology, Brisbane, Australia, under the direction of Prof. B. Boashash. His fields of research include nonlinear dynamical systems theory, nonlinear signal and system modeling and identification, chaos theory and its application, nonlinear adaptive algorithms, and biomedical engineering.

Jean-Marc Vesin received the B.Sc. degree from the Ecole Nationale Supérieure d’Ingénieurs Electriciens de Grenoble (ENSIEG), Grenoble, France, in 1980. He received the M.Sc. degree from the Laval University, Québec City, P.Q., Canada, in 1984, where he spent four years on research projects. After two years in industry, he joined the Signal Laboratory (LTS), Swiss Federal Institute of Technology, Lausanne, Switzerland, from which he received the Ph.D. degree in 1992.

Since then, he has been in charge of activities dealing with one-dimensional signal processing, nonsignal modeling and analysis, and the applications of genetic algorithms in processing.