A Global Least Mean Square Algorithm for Adaptive IIR Filtering

William Edmonson, Jose Principe, Kannan Srinivasan, and Chuan Wang

Abstract—In this brief, we develop an least mean square (LMS) algorithm that converge in a statistical sense to the global minimum of the mean square error (MSE) objective function. This is accomplished by estimating the gradient as a smoothed version of the MSE. The smoothed MSE objective begins as a convex functional in the mean. The amount of dispersion or smoothing is reduced, such that over time it becomes the true MSE as the algorithm converges to the global minimum. We show that this smoothing behavior is approximated by appending a variable noise source to the infinite impulse response (IIR)–LMS algorithm. We show, experimentally, that the proposed method does converge to the global minimum in the cases tested. A performance improvement over the IIR–LMS algorithm and the Steiglitz–McBride algorithm has been achieved.

I. INTRODUCTION

Adaptive filtering represents a major research area in digital signal processing, communications, and control. There exist many applications of adaptive filtering in communications and signal processing that require filters that self-modify, based on the signals encountered within their operating environment. Examples of important applications include linear prediction, adaptive differential pulse coding, echo cancellation, channel equalization, and system identification [6].

Adaptive filters based upon the finite-impulse response (FIR) structure have matured to a point of practical implementations. A major drawback of the adaptive FIR filter is that certain applications will require a very large number of parameters to achieve good performance, thus, increasing computational costs. This becomes evident when the system to be modeled or identified is represented as a pole–zero model.

On the other hand, adaptive filters based upon the infinite-impulse response (IIR) structure [4] have the advantage of approximating a pole–zero model more accurately than the FIR structure. This increased accuracy can be accomplished with an equivalent-order IIR filter, thereby reducing the computational cost in terms of the number of coefficients to be estimated. Although adaptive IIR filters require less coefficients to be estimated, the system may become unstable during adaptation. Another problem area is that the objective function for an adaptive IIR filter can be nonconvex, which implies the existence of multiple local minima. Adaptive IIR filtering typically uses gradient search techniques, e.g., the least mean square (LMS) algorithms [6], which are sensitive to initial conditions. Therefore, these techniques can easily converge to a local minimum, resulting in a suboptimal solution. Hence, adaptive IIR filters are not used commercially at this time. Additional open problems of adaptive IIR filtering, such as convergence to an unstable filter, are discussed in [1] and [2].

We propose to address the problem of convergence to a local minimum of an adaptive IIR filter by investigating the use of stochastic global-optimization methods. This type of global optimization procedure has the property of converging to the global minimum with a probability of one, as the number of iterations tends to infinity [21]. One such general method is the stochastic approximation method, which represents a simple approach to minimizing a nonconvex function. This method is based on using a randomly distributed process to find the absolute minimum of an objective function [3], [10], [16]. In particular, stochastic approximation with convolution smoothing (SAS) has been successfully used as a global optimization algorithm in several applications [5], [11], [12]. Though similar to simulated annealing [13], SAS was empirically proven to be more efficient computationally and more accurate in converging to a global minimum [5]. The objective of convolution smoothing is to "smooth" the nonconvex objective function by convolving it with a noise probability density function (pdf). The variance on the pdf at the start of the optimization procedure is large, which has the effect of "smoothing" the objective function so that it is convex. Then the variance is slowly reduced to zero, whereby the smooth functional returns to the original objective function, as the algorithm converges to the global minimum.

The SAS method represents an off-line procedure for optimizing deterministic objective functions where the data is static and, therefore, is not conducive for adaptive filtering. We will develop an on-line approximation of this method for time-series data. The proposed method is developed from the SAS algorithm by first showing that an on-line version of the algorithm computes the gradient at the present location perturbed by a random value. Secondly, we approximate this gradient by an instantaneous function of its Taylor series expansion. Combining this approximation of the gradient with the LMS algorithm results in a stochastic global optimization algorithm for adaptive IIR filtering. The resultant global optimization LMS algorithm consists of the standard LMS algorithm with the addition of a noise term, whose variance is initially large and approaches zero as the iteration progresses in time. This formulation only incrementally increases the computational cost of the LMS algorithm. Experimentally, we show that the proposed algorithm converges to the global minimum, thereby, alleviating a major problem of adaptive IIR filtering.

This brief is organized as follows: Section II is an overview of the general SAS algorithm, Section III develops the global least-mean-square (GLMS) algorithm for adaptive IIR filtering. Shown in Section IV are the experimental results of using the GLMS method for identifying an unknown system, along with a comparison of its behavior to the IIR–LMS algorithm [6] and the Steiglitz–McBride algorithm [17], and Section V gives concluding remarks.

II. BACKGROUND

SAS is an unconstrained global-optimization algorithm for minimizing a nonconvex function

$$\min_{x \in \mathbb{R}^n} g(x).$$

(1)
The smoothing process represents the convolution of \( g(x) \) with a smoothing function \( \hat{h}(\eta, \beta) \), where \( \eta \in \mathbb{R}^n \) is a random vector used to perturb \( x \), and \( \beta \) controls the degree of smoothing. This smoothed functional, described in [14], is given by

\[
\hat{g}(x, \beta) = \int_{-\infty}^{\infty} \hat{h}(\eta, \beta) g(x - \eta) d\eta
\]

which represents an averaged version of \( g(x) \) weighted by \( \hat{h}(\cdot, \beta) \). To yield a properly-smoothed functional \( \hat{g}(x, \beta) \), the kernel functional \( \hat{h}(\eta, \beta) \) must have the following properties [14]:

1. \( \hat{h}(\eta, \beta) = \left( 1/\beta^n \right) \hat{h}(\eta/\beta) \) is piecewise differentiable w.r.t \( \beta \);
2. \( \lim_{\beta \to 0} \hat{h}(\eta, \beta) = \delta(\eta) \); (\( \delta(\eta) \) is the Dirac delta function);
3. \( \lim_{\eta \to 0} \hat{g}(x, \beta) \equiv g(x) \);
4. \( \hat{h}(\eta, \beta) \) is a pdf.

Under these conditions, we can rewrite (2) as the expectation with respect to \( \eta \)

\[
\hat{g}(x, \beta) = E_\eta [g(x - \eta)].
\]  

In our case, \( \hat{h}(\eta, \beta) \) will be the sampled values of its pdf, which is convolved with the original objective function for smoothing. Both Gaussian, uniform, and Cauchy distributions satisfy the above properties. Throughout this paper, we will be using the Gaussian distribution.

The value of \( \beta \) plays a dominant role in the smoothing process by controlling the variance of \( \hat{h}(\eta, \beta) \); see properties (b) and (c). Furthermore, property (c) states that to avoid convergence to a local minimum, \( \beta \) has to be large at the start of the optimization process and is then gradually reduced to approximately zero as the global minimum is reached. Therefore, there will be a set of smoothed functionals \( \hat{g}(x, \beta) \), \( \beta = 1, 2, \ldots \) that are to be evaluated for different values of \( \beta \) before the global optimum point is reached.

It has been shown in [5] that the formal definition of the SAS optimization problem is to

\[
\min_{x \in \mathbb{R}^n} \hat{g}(x, \beta)
\]  

with \( \beta \to 0 \) as \( x \to x^* \), where \( x^* \) is the global minimum of \( g(x) \). The application of this technique to the LMS algorithm requires a gradient operation on the functional \( \hat{g}(x, \beta) \), i.e., \( \nabla_x \hat{g}(x, \beta) \). As described in [5], [14], if \( \hat{h}(\eta, \beta) \) is a standard multivariate normal distribution, then the unbiased gradient estimate of the smoothed functional when only the gradient of \( g(\cdot) \) is known can be expressed as

\[
\nabla_x \hat{g}(x, \beta) = \frac{1}{N} \sum_{i=1}^{N} \nabla_x g(x - \beta \eta_i).
\]  

The double-sided gradient estimate of (5) can be expressed as

\[
\nabla_x \hat{g}(x, \beta) = \frac{1}{2N} \sum_{i=1}^{N} \left[ \nabla_x g(x + \beta \eta_i) + \nabla_x g(x - \beta \eta_i) \right].
\]  

The SAS algorithm, as presently developed, represents an off-line iterative global optimization algorithm for deterministic objective functions. The next section describes how to adapt the SAS algorithm for use as an on-line procedure for stochastic time series. This entails making the algorithm both iterative and computationally simple by combining the concept of the SAS gradient estimate with the LMS optimization procedure. We define this proposed procedure as the global least mean square (GLMS) algorithm.

III. TIME-SERIES STOCHASTIC APPROXIMATION

To further develop the SAS procedure for use as an adaptive filtering algorithm, we need to place the above development in the context of a stochastic time series. This is accomplished by equating the parameters of the SAS algorithm to that of the general form for adaptive filtering. The general form of this optimization procedure, which has its beginnings with the stochastic approximation procedure of Robbins and Monro [10] and, in particular, Kiefer and Wolfowitz [16] is described as

\[
\theta(n + 1) = \theta(n) + \lambda(n) \nabla \xi(n, \theta) + \gamma^2 \epsilon(n, \theta)
\]  

where \( \theta \) is the parameter vector, \( \lambda(n) \) and \( \gamma \) are gain sequences, and \( \nabla \xi(\cdot) \) is the gradient estimate of the mean square error (MSE), \( \xi(n, \theta) \). The last term on the right-hand side (RHS) represents a noise term to aid in convergence and is usually zero for gradient type algorithms. We will show that the term \( \epsilon(n, \theta) \) develops naturally from the derivation of a time series SAS algorithm as a Gaussian-distributed white noise.

A. Derivation of Gradient Estimate

The key to implementing any algorithm for adaptive filtering (7) is the development of an on-line gradient estimate \( \nabla \xi(n, \theta) \). Here we propose to apply the SAS derived single-sided gradient estimate of (5) to the gradient estimate of (7), i.e.,

\[
\nabla \xi(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \nabla \xi(n, \theta - \beta \eta_i).
\]  

A major principle of the LMS algorithm is computational simplicity and the ability to track the changing characteristics of the time series. We can obtain these attributes by setting \( N = 1 \) in (8), as suggested in [5], yielding the one-sample gradient estimate

\[
\nabla \xi(n, \theta - \beta \eta_i).
\]  

This equation is repeatedly applied for each input sample. The variable \( n \) is added for clarification and denotes the time index in the data.

Conceptually, (9) states that the on-line version of the SAS computes the gradient at a randomly-selected point within a neighborhood of the present operating point. The neighborhood is controlled by \( \beta \), which decreases through the adaptation process. In order to implement (9) on-line, a structure consisting of two filters is necessary, one to compute the input–output map and the other to compute the gradient at the perturbed state \( (\theta - \beta \eta_i) \). In applications where the order of the filter is small, this implementation may be reasonable, but for large-order systems it may be a costly requirement. We investigate below a simplification that involves expressing the gradient at \( \theta - \beta \eta_i \) as a Taylor series around the operating point \( \theta \), i.e.,

\[
\nabla \xi(n, \theta - \beta \eta_i) \approx \left[ \xi(n, \theta) + \beta \xi^\prime(n, \theta) + \frac{(\beta \eta_i)^2}{2!} \xi^{\prime\prime}(n, \theta) + \cdots \right].
\]  

Since the evaluation of the gradient in (10) is done at the operating point \( \theta \), we can use the same filter to compute both the I/O map and the gradient. As a first approximation to (10), we just keep the first two terms and assume a diagonal Hessian. This results in the following gradient estimate that will be used to develop the GLMS algorithm

\[
\nabla \xi(n, \theta - \beta \eta_i) \approx \xi^\prime(n, \theta) - \beta \eta_i.
\]  

The reason for this extreme approximation can be understood from the nature of the gradient estimate dictated by the SAS procedure.
For adaptive IIR filtering, it is important that \( \nabla \xi(n, \theta) \) has the properties for convergence to a global solution that yields a stable filter and is computationally simple. We derive the GLMS algorithm with these properties in mind.

The term \( \xi'(n, \theta) \) of (11) represents the unperturbed gradient of the MSE, which has been previously evaluated for adaptive IIR filtering by Widrow with the IIR–LMS algorithm [6] and by Ljung and Soderstrom with their recursive prediction-error algorithm [8]. Their development leads to the following recursive evaluation of the gradient with respect to the parameters \( \theta \),

\[
\xi'(n, \theta) = \nabla \xi(n, \theta) = \begin{bmatrix}
\nabla \alpha_i(n, \theta) \\
\nabla b_i(n, \theta)
\end{bmatrix}
\]

(19)

The partial derivatives \( \partial \xi(n, \theta) / \partial \alpha_i \), where \( a_i \)'s are the feedback parameters, is evaluated as

\[
\frac{\partial \xi}{\partial \alpha_i(n)} = -\xi(n) \frac{\partial y(n)}{\partial \alpha_i(n)} = -\xi(n) \alpha_i(n), \quad i = 1, \ldots, N - 1 \quad (20)
\]

where

\[
\alpha_i(n) = \sum_{k=1}^{N-1} [a_i \alpha_i(n - k) + y(n - i)].
\]

(21)

Similarly for the feedforward parameters \( b_i \)'s, \( \partial \xi(n) / \partial b_i \) is evaluated as

\[
\frac{\partial \xi(n)}{\partial b_i(n)} = -\xi(n) \frac{\partial y(n)}{\partial b_i(n)} = -\xi(n) \gamma_i(n), \quad i = 1, \ldots, M - 1
\]

(22)

where

\[
\gamma_i(n) = \sum_{k=0}^{M-1} [a_i \gamma_i(n - k) + x(n - i)].
\]

(23)

Combining the gradient-estimation equations of (20) and (22) with (11), and putting them in the appropriate context of the general adaptive algorithm of (7). This results in an on-line stochastic approximation algorithm for adaptive IIR filtering, which we define as the GLMS algorithm

\[
\theta(n + 1) = \theta(n) - \lambda(n) \Xi(n) + \eta \beta.
\]

(24)

The variable \( \Xi(n) \) represents the vector of gradient estimates \( [\xi(n)A, \xi(n)B] \) where \( A = [\alpha_i(n)^T] \) and \( B = [\gamma_i(n)^T] \).

IV. EXPERIMENTAL RESULTS

A. System Identification

In this section, we present experimental results comparing the relative performance of the GLMS algorithm with the Steiglitz–McBride method [20] and the IIR–LMS method [6], using a system identification formulation of the adaptive filter. We will identify the following unknown system

\[
\hat{H}(z) = \frac{0.05 - 0.44z^{-1}}{1 - 1.314z^{-1} + 0.25z^{-2}}
\]

(25)

by the following underparameterized system

\[
\hat{H}(z) = \frac{b}{1 - a z^{-1}}.
\]

(26)

The main goal is to determine the values of the coefficients \( \{a, b\} \) in (26), such that the MSE \( \xi \) is minimized to its global minimum. The input is chosen to be random Gaussian noise with a mean of zero and a variance of one. There exist two minima on the MSE performance...
This is evident from its convergence in the
performance surface has
local minima [23]. Hence,
which approximately represents a discrete exponential function,
whereby each value of $\beta$ is held for 200 iterations.

Our first experiment is to determine the parameters of (26) that
best identifies the system (25) with no observation noise. The step
size $\gamma$ was chosen to be a constant equaling 0.001. Table I shows the
comparison of the sample mean and standard deviation of the GLMS
with the SMM and the IIR–LMS. The results are based on 100 Monte
Carlo (MC) simulations along with the initial condition of
the SMM do not represent convergence to the global minimum
when the desired signal is the output of a
shift-invariant linear system. This method uses the residue theorem,
so it is impractical for on-line operation, but provides the optimal
performance of the filter [23]. Second, the adaptation
of gamma-filter coefficients $W$ and $\mu$ are a parametric least-square
problem, which can be solved iteratively using a gradient-descent
procedure.

We choose the gamma filter for the demonstration of GLMS for two
reasons: First, the performance surface can be analytically computed
for different values of $K$ when the desired signal is the output of a
shift-invariant linear system. This method uses the residue theorem,
so it is impractical for on-line operation, but provides the optimal
value for the coefficient $\mu$ of the filter [23]. Second, the adaptation
of the gamma filter parameters $W$ and $\mu$ represents a nonconvex
optimization problem. Fig 4 depicts a cut of performance surfaces
for different values of the filter order $K$ along the $\mu$-direction when
$W$ is set at its optimal values. The curve for $K = 1$ is convex, but
for $K > 1$ the performance surface has $K$ local minima [23]. Hence,
a gradient search in the space of $W$ and $\mu$ is not guaranteed to find
the global minimum.

A third-order gamma filter was used to identify the following
third-order elliptic filter:

$$H(z) = \frac{0.0563 - 0.0009z^{-1} - 0.0009z^{-2} + 0.0563z^{-3}}{1 - 2.1291z^{-1} + 1.7834z^{-2} - 0.5435z^{-3}}. \quad (27)$$
From Fig. 4 the optimum value of the GLMS algorithm, we will only present the results for Since we are highlighting the global-convergence characteristics of one of the above simulation is shown in Fig. 5. The convergence of the parameter $\mu$ for $K = 3$. The results of this experiment are shown in Table III and correspond very closely to the true optimum value.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Actual</th>
<th>GLMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu$</td>
<td>0.64</td>
<td>0.637±0.0034</td>
</tr>
</tbody>
</table>

Since we are highlighting the global-convergence characteristics of the GLMS algorithm, we will only present the results for $\mu$. This experiment is based on 100 MC simulations with SNR = 10 dB. The schedule for $\beta$ is

$$\beta = \{1, 0.8, 0.5, 0.25, 0.1, 0.05, 0.01, 0.008, 0.003, 0.0007\}.$$

From Fig. 4 the optimum value of $\mu$ is 0.64 for $K = 3$. The results of this experiment are shown in Table III and correspond very closely to the true optimum value. The convergence of the parameter $\mu$ for one of the above simulation is shown in Fig. 5.
A Compact Neural-Network-Based CDMA Receiver

David C. Chen and Bing J. Sheu

Abstract—A compact neural-network receiver for code-division multiple-access (CDMA) communication is presented. In the CDMA scheme, the near–far problem is a major impediment for the performance of a conventional detector (CD). By using a compact neural network with hardware annealing function and gradient-descent optimization technique, an optimal multiuser receiver can be implemented. The algorithm and simulation results of the compact neural-network-based CDMA receiver are described in this brief.

Index Terms—Optimization, multimedia, neural network, receiver, spread spectrum, wireless communication.

I. INTRODUCTION

Code-division multiple-access (CDMA) communication makes use of spread-spectrum technology. It has received significant attention for digital mobile telephone and personal communication systems (PCS’s). In CDMA communication, each user is assigned a unique and distinctive code. These codes are almost uncorrelated and used to spread the transmitted signals to the full available bandwidth. Thus, signal collisions are not destructive and each of the signals involved in a collision only results in a slight increase in the error rate. Many subscribers are allowed to share the same frequency band, the capacity is increased, and the requirement of allocating different frequencies to different users is eliminated.

To ensure high transmission quality, it is very important to control the signal power quickly and accurately. However, this is not an easy task. In satellite communication, the high- and low-power transmitters coexist. In ground communication, one user may be close to the receiver while the other user may be far from it. When an unwanted user’s received signal power is much larger than the received signal power contributed by the desired user, the performance of the CDMA system is significantly degraded. This problem is referred to as the near–far problem and is one major technical challenge in the CDMA system. To achieve high communication quality, it is necessary to develop a reliable technology to reduce or overcome the effects of the near–far problem.

Recently, Aazhang et al. [1] and Mitra et al. [2] proposed feedforward neural-network-based receivers for multiuser accessing. They achieved good performance when the number of users was small. However, the associated hardware complexity will increase exponentially as the number of users is increased. In 1986, Verdu [3] reported that an optimized receiver for near–far resistant multuser demodulation was possible by minimizing a quadratic objective function, assuming that users’ signals were uncorrelated and their spreading codes, i.e., pseudorandom codes, were known. Kechronis [4] proposed a receiver based on the Hopfield neural network, but the optimal solution was not guaranteed. Our proposed compact neural network with the hardware annealing optimization is capable of finding the optimized solution by minimizing a quadratic objective function. In this brief, a receiver based on a compact neural network for the CDMA system with robust resistance to the near–far problem is presented. The optimized solution can be achieved quickly and the required circuit complexity is moderate with the use of regular circuit components.

II. OPTIMIZED DECISION RULE OF CDMA RECEIVER

For illustrative purposes, we assume $K$ active users for the same synchronous Gaussian channel in a CDMA system at a given time $t$. There are $K$ different signature waveforms $\{s_k(t), k = 1, 2, \ldots, K\}$. Each $s_k$ is composed of a string of bits $[b_k(i) \in \{-1, +1\}]$. In a CDMA system, the received signal is

$$r(t) = \sum_{k=1}^{K} b_k(i) s_k(t - iT) + n(t), \quad t \in [iT, (i + 1)T]$$

(1)

where $n(t)$ is Gaussian noise. If we focus on one symbol interval in (1), the function of a receiver is to recognize every active user’s symbol at the specified interval. There are two kinds of detectors used in a receiver. One is called the conventional detector (CD), and another is the optimal multiuser detector (OMD).

A. Conventional Detector

A CD consists of $K$ filters matched to the signature waveforms of $K$ subscribers. Simple decision devices following the matched filters provide every user’s symbol estimates based upon the signs of the output of the matched filters at the specific time interval:

$$y_k^{(i)} = \int_{iT}^{(i+1)T} r(t) s_k(t - iT) \, dt.$$

$$b_{CD}^{(i)} = \text{sign}(y_k^{(i)})$$

(2)

where $b_{CD}^{(i)} \equiv [b_1^{(i)}, b_2^{(i)}, \ldots, b_K^{(i)}]$ and $y^{(i)} \equiv [y_1^{(i)}, y_2^{(i)}, \ldots, y_K^{(i)}]^T$. The block diagram of a CD is shown in Fig. 1.

B. Optimal Multiuser Detector

The OMD can be constructed by minimizing the noise energy, which is equivalent to maximizing the logarithm of the likelihood function. The matrix representation of $y$ is

$$y = \mathbf{H} \mathbf{b} + \mathbf{n}$$

(3)

where $\mathbf{n} = [n_1^{(i)}, n_2^{(i)}, \ldots, n_K^{(i)}]^T$ and $\mathbf{H} \in R^{K \times K}$ is a cross-correlation matrix of the signature waveforms

$$h_{ij} = \int_0^T s_i(t) \cdot s_j(t) \, dt.$$  

(4)