ALGORITHMIC ANALYSIS AND IMPLEMENTATION OF A NOVEL NATURAL GRADIENT ADAPTIVE FILTER FOR SPARSE SYSTEMS

Finbarr O'Regan and Conor Heneghan

Digital Signal Processing Laboratory,
Department of Electronic and Electrical Engineering,
University College Dublin,
Dublin 4, Ireland
E-mail: finbarr.oregan@ee.ucd.ie, conor.heneghan@ucd.ie

ABSTRACT

We present analytical results, and details of implementation for a novel adaptive filter incorporating an approximate natural gradient tap-update algorithm, termed the simplified signed sparse LMS algorithm (SSSLMS). Each tap-update equation includes a term proportional to the tap-value, so that larger taps adapt more quickly than for a corresponding Least Mean Square (LMS) update. Results indicate that the algorithm is suited for use in sparse channels. The bounds on its maximum allowable step-size differ from LMS, and simulations are provided that indicate potentially more robust convergence for larger step-sizes than LMS. A theoretical expression for the excess mean square error (MSE) is also derived, and confirmed by numerical simulation. Fixed point simulations of the algorithm using a proposed hardware architecture are also presented. The computational complexity is of the same order as the standard LMS. Finally, profiling of the power consumption of the SSSLMS implementation indicate that the architecture consumes approximately twice as much power as a standard LMS implementation.

1. INTRODUCTION

Adaptive filters are at the core of many modern communication systems, and find common use as equalizers, echo cancelers, and crosstalk cancelers, to name but a few applications. In these cases, the target for the adaptive filter is often a *sparse* finite impulse response (FIR) filter. Loosely speaking, a sparse channel can be defined as a long channel with many relatively insignificant taps and a few significant taps. Alternatively, it can be described as a filter where the probability density function for tap weights is heavily skewed towards zero values. Another interpretation of sparse filters is to consider them as non-uniformly spaced filters. In filters adapting to sparse channels, low magnitude taps do not contribute significantly towards reducing the mean square error between desired and actual outputs; however they can be a significant consumer of power, as their weights still need to be continuously updated. In addition, the need to adapt to low magnitude taps contributes to overall convergence time with negligible benefit in residual error.

As signaling speeds increase in data communications, long sparse channels are becoming more commonly encountered, while conversely the desire for low power applications requires these long filters to minimize power consumption. This has led to efforts to consider alternatives to the standard Least Mean Squares (LMS) specifically adapted to sparse channels, with the goals of reducing convergence time, reducing excess mean square error, reducing complexity and/or power, or combinations of these properties.

In this paper, we consider an alternative to the well known LMS algorithm for weight updates. This algorithm is termed a simplified signed sparse LMS (SSSLMS) algorithm, and can provide faster convergence for sparse channels under certain circumstances. This algorithm is a special case of the more general class of adaptive natural gradient algorithms proposed in [1], and a similar algorithm was studied by Martin *et al.* [2]. Our algorithm differs from Martin's signed sparse LMS in that it allows sign changes in the tap weights by reverting to the standard LMS update. The hardware implementation is presented in section 3.

2. ANALYSIS OF THE SIGNED SPARSE LMS ALGORITHM

2.1. The Weight Update Equation

The conventional LMS algorithm has the well-known weight update equation given by

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu e_n \mathbf{x}_n, \tag{1}$$

where $\mathbf{w}_n$ is the tap-weight vector of the adaptive filter at the n^{th} interval, $\mathbf{x}_n$ is a regressor vector of the current and previous inputs to the adaptive filter, μ is a step size parameter, and $e_n = d_n - \mathbf{w}_n^T \mathbf{x}_n$ is the error between actual and desired outputs (where d_n is the desired response). In contrast, the exponentiated gradient (EG) algorithm first developed by Kivinen and Warmuth [3] uses a weight update equation

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu e_n \left(\mathbf{W}_n + \epsilon^2 \mathbf{I}\right) \mathbf{x}_n, \tag{2}$$

where, if L is the length of the adaptive filter then $\mathbf{W}_n = \text{diag}\left\{ \left|w_n^{(0)}\right| \quad \left|w_n^{(1)}\right| \quad \ldots \quad \left|w_n^{(L-1)}\right| \right\}$ is a $L \times L$ diagonal matrix with entries equal to the magnitude of the tap weights, and $\mathbf{I}$ is the identity matrix. The purpose of the ϵ^2 term is to ensure that the update term is not equal to zero in the case where $w_n^{(i)}$ is at zero. The presence of the ϵ^2 term also allows the weights to transition across the zero boundary. (Without this term, the update term would become smaller and smaller as the true tap weight approached zero, and for small step size it would be extremely unlikely that the tap weight would change sign). However, the ϵ^2 term increases the complexity of hardware implementation, and also contributes to excess mean square error. Note also that in the case where the current $w_n^{(i)}$ has the opposite sign as compared to its optimum Wiener weight, then it will tend to train to a zero-magnitude tap. An interpretation of this algorithm is that it is similar to the LMS weight update algorithm, except that each tap has effectively a separate step size μ with

$$\boldsymbol{\mu}_{\text{eg}} = \mu_{\text{lms}} \left| \mathbf{w}_n \right| . \tag{3}$$

The potential advantage of this EG algorithm is that the magnitude of the update term for each tap weight is proportional to the tap weight itself. Therefore intuitively, high magnitude taps should converge more quickly due to their higher effective step size, whereas small-magnitude taps should contribute less excess MSE than when the conventional LMS is used. This insight led Martin *et al.* to rename equation (2) as a signed-sparse LMS (SSLMS), since such properties seem beneficial to sparse system identification.

We propose a modified SSLMS algorithm, referred to as the SSSLMS, in which the ϵ^2 term is discarded. Therefore, for small step size it is extremely unlikely that the tap weights will change sign. We maximize the probability of the tap weights having the correct sign and also allowing taps to cross the zero boundary by alternating the update between the SSSLMS and the standard LMS algorithms.

For our SSSLMS algorithm, the weight update equation is simply

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu e_n \mathbf{W}_n \mathbf{x}_n . \tag{4}$$

The architecture proposed in Section 3 in fact allows an alternating update procedure similar to the PNLMS++ in [4] to overcome.

2.2. Convergence and Excess Mean Square Error of SSSLMS Algorithm

We start the analysis by defining a weight error vector as

$$\boldsymbol{\varepsilon}_n = \mathbf{w}_n - \mathbf{w}_o . \tag{5}$$

Equation (4) can then be expressed as

$$\begin{aligned}
\boldsymbol{\varepsilon}_{n+1} &= \boldsymbol{\varepsilon}_n + \mu \mathbf{W}_n \mathbf{x}_n e_n & (6)\\
&= \boldsymbol{\varepsilon}_n + \mu \mathbf{W}_n \mathbf{x}_n \left(d_n - \mathbf{x}_n^T \left(\boldsymbol{\varepsilon}_n + \mathbf{w}_o \right) \right) & (7)\\
&= \boldsymbol{\varepsilon}_n - \mu \mathbf{W}_n \mathbf{x}_n \mathbf{x}_n^T \boldsymbol{\varepsilon}_n + \mu \mathbf{W}_n \mathbf{x}_n e_n^* & (8)
\end{aligned}$$

where $\mathbf{w}_o$ is the optimum Wiener filter, and the optimum error is given by $e_n^* = d_n - \mathbf{x}_n^T \mathbf{w}_o$. Using a direct averaging method [5], under the assumption that μ is small, this stochastic difference equation is similar to [4]

$$\boldsymbol{\varepsilon}_{n+1} = \boldsymbol{\varepsilon}_n - \mu E\left\{ \mathbf{W}_n \mathbf{x}_n \mathbf{x}_n^T \right\} \boldsymbol{\varepsilon}_n + \mu E\left\{ \mathbf{W}_n e_n^* \mathbf{x}_n \right\} . \tag{9}$$

In the limit as $n \to \infty$, $E\{\mathbf{W}_n\} = \mathbf{W}_o$ (a diagonal matrix with the optimum Wiener weight magnitudes along the diagonal). Using independence assumptions [5], equation (9) becomes

$$\boldsymbol{\varepsilon}_{n+1} = \boldsymbol{\varepsilon}_n - \mu \mathbf{W}_o \mathbf{R} \boldsymbol{\varepsilon}_n + \mu \mathbf{W}_o \mathbf{x}_n e_n^* \tag{10}$$

with $\mathbf{R} = E\left\{ \mathbf{x}_n \mathbf{x}_n^T \right\}$.

The correlation matrix of the weight-error vector $\boldsymbol{\varepsilon}_n$ is defined as $\mathbf{K}_n = E\left\{ \boldsymbol{\varepsilon}_n \boldsymbol{\varepsilon}_n^T \right\}$. Under further independence assumptions and using the definition of $\mathbf{K}_n$, we can write

$$\begin{aligned}
\mathbf{K}_{n+1} = &\left(\mathbf{I} - \mu \mathbf{W}_o \mathbf{R} \right) \mathbf{K}_n \left(\mathbf{I} - \mu \mathbf{W}_o \mathbf{R} \right)^T \\
&+ \mu^2 J_{\text{min}} \mathbf{W}_o \mathbf{R} \mathbf{W}_o & (11)
\end{aligned}$$

where J_{min} is the minimum mean square error. Since all matrices in equation (11) are positive definite for small μ, a convergent solution satisfies:

$$\begin{aligned}
\mathbf{K} = &\left(\mathbf{I} - \mu \mathbf{W}_o \mathbf{R} \right) \mathbf{K} \left(\mathbf{I} - \mu \mathbf{W}_o \mathbf{R} \right)^T \\
&+ \mu^2 J_{\text{min}} \mathbf{W}_o \mathbf{R} \mathbf{W}_o & (12)
\end{aligned}$$

The solution to this is given by [2]:

$$\mathbf{K} = \frac{1}{2} \mu J_{\text{min}} \mathbf{W}_o \left(\mathbf{I} - \frac{\mu}{2} \mathbf{R} \mathbf{W}_o \right)^{-1} . \tag{13}$$

This is only guaranteed to be positive definite if all eigenvalues of $\frac{\mu}{2} \mathbf{R} \mathbf{W}_o$ are less than 1. Therefore convergence conditions for the SSSLMS algorithm are

$$0 \leq \mu \leq \frac{2}{\lambda_{\text{max}}} , \tag{14}$$

where λ_{max} is the maximum eigenvalue of the matrix $\mathbf{R} \mathbf{W}_o$.

For such a convergent solution, an expression for the excess square error is given by [5]:

$$J_{\text{ex}}(\boldsymbol{\varepsilon}_n) = \text{trace}\left\{ \mathbf{R} \mathbf{K} \right\} \tag{15}$$

Under simulation conditions where $\mathbf{R}$ and $\mathbf{W}_o$ are known, the excess MSE can be numerically evaluated. Since an expression for the excess MSE of the LMS algorithm is well known, the convergence properties of both algorithms can be fairly compared by selecting both to have the same excess MSE.

3. HARDWARE IMPLEMENTATION OF THE SSSLMS ALGORITHM

3.1. Hardware Architecture

Figure 1 shows the proposed architecture for a single tap section of the SSSLMS algorithm. This circuit implements equation (4) using two's-complement signed arithmetic. There are two multiplexers in the weight update block, labeled as B and D. Multiplexer B implements the $|\mathbf{W_n}|$ functionality using a ones complement inversion. A control signal, *revert_to_lms* is used allow the

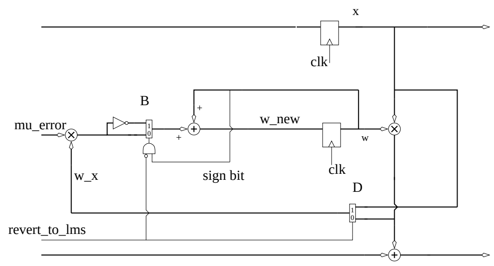

Fig. 1. Hardware architecture for SSSLMS algorithm.

architecture to operate using the standard LMS algorithm by controlling multiplexer D (and necessarily multiplexer B). This allows an easy swap between the LMS algorithm and the SSSLMS algorithm modes. By alternating the update between LMS mode and SSSLMS mode, we ensure that taps train to their correct weights (similar to the alternating update equation suggested in ([4]). The alternating update mechanism also means that the taps cannot lock at a zero value. This architecture is simpler than would be necessary for the PNLMS [6] and greatly simpler than the improved PNLMS suggested by Benesty and Gay in [7].

4. RESULTS

4.1. Infinite Precision Results

The analysis of Section 2 predicts several ways in which the behavior of the SSSLMS algorithm differs from that of conventional LMS.

Firstly, the analysis predicts that the SSSLMS algorithm can have a higher μ_{critical} where μ_{critical} is the largest possible step size defined by equation (14). This is because the eigenvalues of $\mathbf{RW}_0$ depend on both the channel and the input sequence, and in certain cases can give a smaller maximum eigenvalue than that of $\mathbf{R}$ alone. To verify this, we conducted simulations using an eleven tap sparse channel (with only three non-zero significant taps) *h=[3 1 3 0 0 0 0 0 0 0 0]*. The adaptive filter was set up in a system-identification configuration. The input signal was white noise with variance equal to 0.001, and both the adaptive filter and the true channel were assigned to be eleven-tap FIR filters. Figure 2 shows the ensemble average mean square error over 200 simulations of the instantaneous squared error for both algorithms. It shows that for $\mu = 0.14$, the LMS algorithm does not converge (though the generally accepted constraint for μ_{lms} to be less than $2/\lambda_{\text{max}}$ is met) while the SSSLMS still achieves an ensemble average MSE of -18dB, thereby confirming the predicted variation in training performance.

Secondly, we confirmed the expression for excess MSE given by equation (15). A similar configuration to that described above was used, but with a channel given by *h=[0.1 -3 3 0.1 -0.1 0.1 -0.2 3.2 0.2 -0.3 0.1]*. The theoretical value of excess MSE for the SSSLMS algorithm was confirmed using an ensemble average of 200 simulations, as shown in Figure 3.

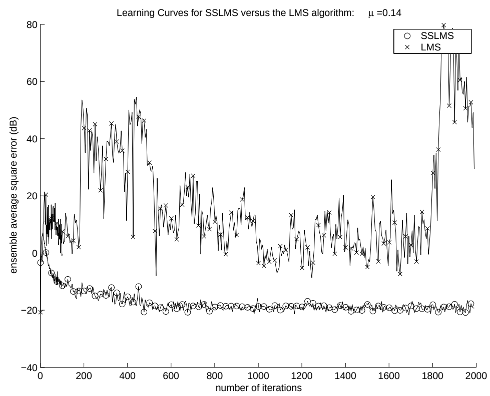

Fig. 2. SSSLMS versus LMS convergence for $\mu_{\text{lms}} = \mu_{\text{ssslms}} = 0.14$, indicating that the SSSLMS can potentially have a larger possible step size than LMS.

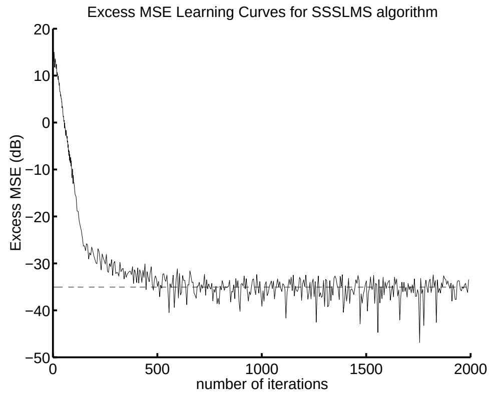

Fig. 3. Ensemble excess MSE for the SSSLMS algorithm, with theoretical steady state excess MSE shown as a dotted line.

4.2. Finite Precision Results

To explore further the convergence rates of the LMS and SSSLMS algorithms, we compared the misalignment of two 64 tap fixed point implementations (written in Verilog). The misalignment is defined as $||\mathbf{w}_o - \mathbf{w}_n||/||\mathbf{w}_o||$. The input data was five level PAM symbols and the tapped delay line filter uses 12 bits for each register in the tapped delay line filter. The weight update blocks use a 16 bit data path to maintain numerical accuracy. The tuning of μ_{ssslms} was critical to the performance of the SSSLMS architecture (a conclusion also identified in [6]). The result shown in Figure (4) indicates that the SSSLMS algorithm can achieve a better misalignment when the channel is dispersive. This is as a result of a reduced effective step size (equation (3) in the case where the tap weights are small. Figure (5) shows how the SSSLMS trains faster than the LMS algorithm in the case of a sparse channel but the misalignment is larger. These results confirm that Adaptive Natural Gradient Algorithms like SSLMS and PNLMS are indeed suited to sparse channels if convergence is the main criterion.

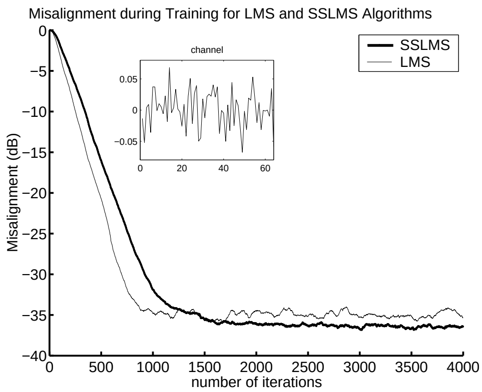

Fig. 4. Misalignment Comparison for a dispersive channel

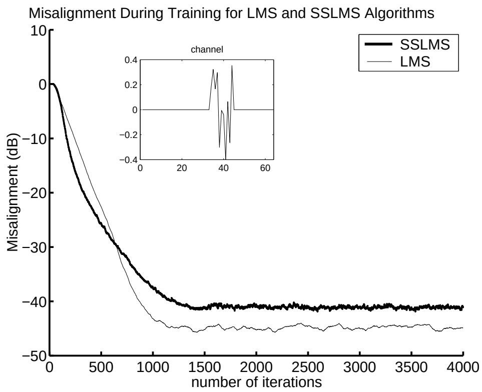

Fig. 5. Misalignment Comparison for a quasi-sparse channel

4.3. Power Consumption Results

We also profiled the power consumption of the architecture using an Verilog power analysis tool[1]. The 64 tap filter was allowed to train in LMS mode and then switched to SSSLMS mode after 1000 symbols. Power consumption in DSP architectures is a function of the filter architecture, data statistics and data activity, number system, arithmetic operator implementation and CMOS technology. This experiment results are shown in Figure (6) and indicate that the larger variance of the weights in steady state SSSLMS mode contribute to activity in both multipliers and hence increases the total dynamic power consumption relative to the LMS algorithm.

5. DISCUSSION

In conclusion, our results show that the SSSLMS algorithm is a possible alternative to the standard LMS algorithm. A potential advantage of the SSSLMS algorithm is that in some circumstances it allows higher values of step-size than the corresponding LMS implementation. It is suited to sparse channels and can be used if the channel is known *a priori* to be sparse e.g acoustic echo and crosstalk channels. If used in a conjunction with a gear-shifting algorithm, this may allow faster convergence. In general, however, if both algorithms are constrained to produce the same excess MSE, the SSSLMS is not guaranteed to converge faster than LMS; the relative convergence speed is a function of both the channel and chosen initial values. We have also given a novel hardware implementation of the SSSLMS algorithm, which incorporates the facility for reversion to the standard LMS. Based on the power profiling experiment we conclude that natural gradient algorithms may not be a promising solution for low power operation.

[1]PowerTheater by Sequence Design

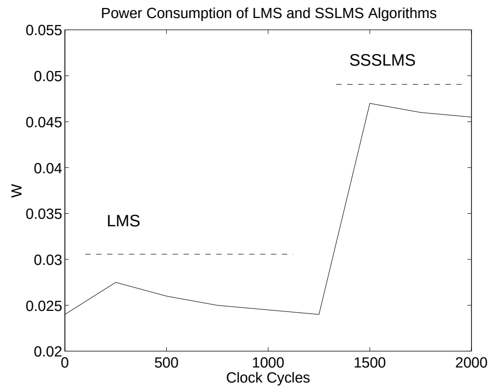

Fig. 6. Power consumption during LMS and SSSLMS training.

6. REFERENCES

[1] S. Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, no. 2, pp. 251–276, Feb. 1998.

[2] R. K. Martin, W. A. Sethares, R. C. Williamson, and C. R Johnson Jr., "Exploiting sparsity in adaptive filters," *IEEE Transactions on Signal Processing*, vol. 50, no. 8, pp. 1883–1894, August 2002.

[3] J. Kivinen and M. K. Warmuth, "Exponentiated gradient versus gradient descent for linear predictors," *Information and Computation*, vol. 132, no. 1, pp. 1–64, Jan. 1997.

[4] S. L. Gay, "An efficient, fast converging adative filter for network echo cancellation," in *Proceedings of the Asilomar Conference*, 1998.

[5] S. Haykin, *Adaptive Filter Theory*, Information and System Science Series. Prentice Hall, 1996.

[6] D. L. Duttweiler, "Proportionate normalized least mean squares adaptation in echo cancelers," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 5, pp. 508–518, September 2000.

[7] Jacob Benesty and Steven L. Gay, "An improved PNLMS algorithm," in *Proceedings of ICASSP*, 2002.