

THE LOG-LOG LMS ALGORITHM

Shivaling S. Mahant-Shetti, Srinath Hosur and Alan Gatherer

DSP R & D Center, Texas Instruments Inc., Dallas, Tx 75265

ABSTRACT

This paper describes a new variant of the least-mean-squares (LMS) algorithm, with low computational complexity, for updating an adaptive filter. The reduction in complexity is obtained by using values of the input data and the output error, quantized to the nearest power of two, to compute the gradient. This eliminates the need for multipliers or shifters in the algorithm's update section. The quantization itself is efficiently realizable in hardware. The filtering section is unchanged. Thus, this algorithm is similar to the sign based variants of the LMS algorithm. However, the complexity of the proposed algorithm is lower than that of the sign-error LMS algorithm, while its performance is superior to this algorithm. In particular, it is close to that of the regular LMS algorithm. The new algorithm also requires much lower area for ASIC implementation.

1. INTRODUCTION

Adaptive filters are widely used in a variety of applications. In all applications, the computational complexity as well as the rate of convergence and steady state mean-square-error (MSE) behavior of the adaptive algorithm are important considerations. The LMS algorithm is used in most cases due to its computational simplicity. In particular, for a filter of length N , the complex LMS algorithm requires $4N$ real multiplications and $4N$ real additions at the input word-length resolution for coefficient update. In many applications including high speed modems and wireless mobile communications, further reduction in complexity is required. The reason for this is reduction in complexity usually leads to lower power consumption and low silicon area. Low power consumption is a key issue in many applications such as wireless communications and lower silicon area is also required in many adaptive equalization applications, where typically, equalizers can occupy up to 50% of the demodulator chip area [1].

The sign based variants of the LMS algorithm, namely, the sign-error, sign-data and sign-sign LMS algorithms have been proposed to reduce complexity [2, 3]. These algorithms *do not* change the filtering section of the adaptive filter. The sign-error LMS algorithm uses the sign of the error while using full (input word-length) resolution data. On the other hand, the sign-data LMS algorithm uses the sign of the data and full resolution error to update the coefficients. Thus, in its most basic form, the complex sign-error LMS algorithm requires $2N$ shifters (for multiplication by the step-size which is chosen as a power of two) and $4N$ full word-length additions for coefficient update. The sign-data LMS algorithm requires only one shifter but still requires

$4N$ full word-length additions. However, the excess MSE of the sign-error LMS is much higher than the regular LMS algorithm. Lowering the step-size to reduce the excess MSE slows the convergence speed. Also, a lot of chip area is still required for interconnect routing for both algorithms. The sign-sign LMS algorithm eliminates the need for shifters and much of the routing but further worsens the convergence and MSE performance. Also, it is known to diverge in situations when the regular LMS algorithm does not [4].

In this paper we propose a novel algorithm for updating the filter coefficients. The algorithm quantizes the input data and the error to the nearest power of two and uses these quantized values to update the adaptive filter coefficients. Like the sign LMS algorithms, the filtering section is left unchanged. The quantization itself is achieved very easily in practice due to the binary representation of inputs. As the quantized values are powers of two, they can be represented in their $\log_2$ form. This implies that these values have a smaller word-length requirements. The complex version of this algorithm requires $2N$ additions at full (input) word-length resolution and $4N+2$ additions at a resolution of $\log_2 L$ per update, where L is the input word-length. In practice the complexity (in terms of area and power) of these reduced word-length additions are equivalent to kN full word-length additions, where k is close to 1. However, the speed of this circuitry is still better, allowing clocking at higher rates. Note that this complexity also includes the circuitry needed to "code" and "decode", i.e., convert the inputs to $\log_2$ numbers and back. Thus, the proposed algorithm's complexity per update of $(2+k)N$, k close to 1, additions is lower than that of the sign-error ($4N$ additions + $2N$ shifts) and sign-data LMS algorithm ($4N$ additions + 2 shifts). We show using simulations, however, that the convergence and MSE performance of the new algorithm is close to that of the regular LMS algorithm. A comparison of the chip area required by this algorithm to that of the sign based LMS variants (see Table 1) also shows that the proposed algorithm requires lower area for ASIC implementation.

2. LOW COMPLEXITY LOG-LOG LMS ALGORITHM

Let the $M \times 1$ input data vector be $\mathbf{x}(n) = [x(n), x(n-1), \dots, x(n-N+1)]^T$. The output error of the adaptive filter is given as $e(n) = d(n) - \mathbf{h}^H(n)\mathbf{x}(n)$, where $d(n)$ is the desired signal, $\mathbf{h}(n)$ is the adaptive filter coefficient vector and $(\cdot)^H$ denotes the Hermitian transpose. The regular LMS adaptation is given as

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \mu e^*(n)\mathbf{x}(n), \quad (1)$$

where μ is the step-size.

The sign based LMS algorithms have been originally proposed for real inputs. However, they can be very easily extended to the complex case by taking signs of the real and the imaginary components independently. The generalized LMS update which uses some functionals of the input data and error to compute the gradient is given as

$$h(n+1) = h(n) + \mu f(e^*(n))g(x(n)). \quad (2)$$

In case of the sign-error LMS algorithm, $f(\cdot)$ is the *sign* function and $g(\cdot)$ is an identity function. For the sign-data LMS algorithm, $f(\cdot)$ is the identity function and $g(\cdot)$ is the *sign* function, while both are chosen to be *sign* functions for the sign-sign LMS algorithm. The function $sign(y)$ for a complex y is defined using the real and imaginary parts of y , y^R and y^I , respectively as $sign(y) = sign(y^R) + j*sign(y^I)$. If the step size μ is chosen to be a power of two, it can be easily seen that the sign-error and the sign-data LMS algorithms require $4N$ full word-length additions per update. Also, the sign-error LMS algorithm requires $2N$ shifts per update. However, neither of the techniques mimic the LMS algorithm closely [3]. The sign-sign algorithm requires only $O(2N)$ adders and no shifters but is known to have a poorer performance. In fact, it is known to diverge in situations where the LMS algorithm is known to converge [4]. We note that the $sign(\cdot)$ function is a form of quantization where the word-length is one bit. Thus, the poor performance of the sign LMS algorithms is due to large amount of quantization noise in the gradient estimate. We can of course increase the resolution of this quantization. However, this would increase the computational complexity by introducing multipliers.

In this paper we propose a new algorithm which uses the values of the error and the data, quantized to the nearest power of two. We show that this new log-log LMS algorithm (so called as it uses a $\log_2$ representation of the quantized error and data) can be implemented at a complexity much lower than that of the sign-error LMS algorithm.

Let, $Q(y)$ represent a power of two value which is closest to y . Again, for a complex y , $Q(y) = Q(y^R) + Q(y^I)$, i.e., the quantization is performed independently on the real and imaginary parts. As mentioned earlier, this quantization is implementable in practise as the inputs to the adaptive filter are usually sampled and quantized using uniform analog-to-digital converters (ADCs). Thus, y already has a binary representation. The LMS adaptation on the real and imaginary parts of the k^{th} tap coefficient, h_k using these quantized values can be written as

$$h_k^R(n+1) = h_k^R(n) + Q(\mu e^R(n))Q(x_k^R(n)) + Q(\mu e^I(n))Q(x_k^I(n)) \quad (3)$$

$$h_k^I(n+1) = h_k^I(n) + Q(\mu e^R(n))Q(x_k^I(n)) - Q(\mu e^I(n))Q(x_k^R(n)). \quad (4)$$

The values $Q(x_k^R(n))$, $Q(x_k^I(n))$, $Q(\mu e^R(n))$ and $Q(\mu e^I(n))$ are all powers of two. Therefore, they can be represented in the " $\log_2$ " domain using fewer number of bits (smaller word-length). This is easily achieved by storing only their corresponding signs and the exponent values. This would

imply very short word-lengths for this type of representation. For example, the nearest power of two quantized representation of a number with a word-length of 32 bits in the $\log_2$ domain requires only 5 bits for the magnitude and one for the sign. Also, the multiplications in equations (3) and (4) can be replaced by an ex-or of the corresponding signs and adding the corresponding exponents. The addition of these exponents is performed at a much lower word-length, leading to savings in both power and silicon. Note also that when μ is a power of two, we need only two low word-length additions ($\log_2$ of the input word-length) to obtain $Q(\mu e(n))$, eliminating the need for the shifters required by the sign-error or the sign-data LMS algorithms. It is easy to see that the correction terms in the above update equation, $Q(\mu e^R(n))Q(x_k^R(n)) + Q(\mu e^I(n))Q(x_k^I(n))$ and $Q(\mu e^R(n))Q(x_k^I(n)) - Q(\mu e^I(n))Q(x_k^R(n))$, represent binary words at the coefficient word-length containing at most two non-zero bits. Thus this update can be implemented very easily in practice.

2.1. Modified Log-Log LMS

We noted above that the correction terms in the log-log LMS update equations of (3) and (4) comprise of binary words which have at most two non-zero bits. For the real case, it is easy to verify that the correction term in the update equation is a binary word which has at most only one non-zero bit. Thus, in order to further reduce the complexity, we can modify the log-log LMS update equations as,

$$h_k^R(n+1) = h_k^R(n) + Q(Q(\mu e^R(n))Q(x_k^R(n)) + Q(\mu e^I(n))Q(x_k^I(n))) \quad (5)$$

$$h_k^I(n+1) = h_k^I(n) + (Q(\mu e^R(n))Q(x_k^I(n)) - (Q(\mu e^I(n))Q(x_k^R(n)))) \quad (6)$$

We can easily see that reduction in complexity is achieved as each of the correction terms in the above update equations is a binary word with at most one non-zero bit. The quantization of the correction terms can be performed very easily as their sub-terms are all powers of two. One easy, though not exact, way of implementing this quantization is to replace $Q(Q(\mu e^R(n))Q(x_k^R(n)) + Q(\mu e^I(n))Q(x_k^I(n)))$ and $Q(Q(\mu e^R(n))Q(x_k^I(n)) - Q(\mu e^I(n))Q(x_k^R(n)))$ in equations (5) and (6) by the larger (w.r.t magnitude) of their corresponding sub-terms. The case where the sub-terms are of equal magnitude can be handled appropriately.

We shall now estimate the complexity of the modified log-log LMS algorithm. The complexity of the log-log LMS algorithm is only slightly higher and comparable to this algorithm. As the complexity of each comparison is comparable to that of an addition, the resulting update algorithm requires only $O(4N + 2)$ low word-length additions for computing the gradient. Note again that these additions are performed at a much lower word-length of $\log_2$ of the original word-length. In practice the complexity (in terms of area and power) of these reduced word-length additions are equivalent to kN full word-length additions, where k is close to 1. However, the speed of this circuitry is still better,

Algorithm	Area
Regular LMS	$N(4L + 5L^2)$
Sign-error LMS	$N(4L + 8L)$
Sign-data LMS	$4NL + 8L$
Sign-sign LMS	$2NL + 2L$
Log-Log LMS	$N(2L + kL), k < 2$ (close to 1 in practice).

Table 1: Comparison of chip area requirements

allowing clocking at higher rates. Note that this complexity also includes the circuitry needed to convert the inputs to $\log_2$ numbers and back. Thus, for typical applications, the complexity of the algorithm is equivalent to $4N + kN$ full word-length additions, where k is close to 1. Thus, we obtain savings both in area and power. In comparison, the sign-error LMS algorithm requires $4N$ full word-length additions and $2N$ shifters while the sign-data LMS algorithm requires only two shifters but still needs $4N$ adders. Thus the resulting modified log-log LMS algorithm results in a savings in complexity with a improvement in MSE and convergence performance over the sign-error algorithm. In particular, its performance is close to that of the LMS algorithm.

The silicon areas of multipliers, adders, shifters and other circuits is proportional to the word-length L . Table 1 compares the areas required by the regular, the sign-error, the sign-data, the sign-sign and the modified log-log LMS algorithm update sections. Note that the complexity of the log-log LMS is only slightly higher than that of the modified log-log LMS algorithm. The area estimates in this table are derived from the number of multipliers, adders etc. required by each algorithm and does not take into account area required for routing. The table shows that the chip area required by the modified log-log LMS is much lower than the sign-error LMS algorithm. This coupled with the fact that interconnect routing for implementing the modified log-log LMS algorithm is a minimum implies a huge savings in area.

3. RESULTS AND DISCUSSIONS

Extensive simulations assessed the performance of the proposed algorithm. The simulations showed that the performance of the proposed algorithm is superior to the sign based algorithms and close to that of the full-LMS algorithm.

We illustrate the performance of our algorithms with a representative example of channel equalization for cable modems. The input to the adaptive filter consisted of a 256 QAM signal. The channel to be equalized represents a well terminated 6MHz downstream cable channel with a roll off of 1dB. The channel frequency response is shown in Fig. 1. The learning curves of the log-log LMS, modified log-log LMS, the sign-error LMS, the sign-data LMS, the sign-sign LMS and the regular LMS algorithms are plotted in Fig. 2. The step size μ was chosen to be 2^{-11} for all algorithms. A linear equalizer with filter length 10 was used in all cases. The input signal to noise ratio was kept at 40dB, which is typical for cable modem applications. The algorithms were implemented in fixed point. All al-

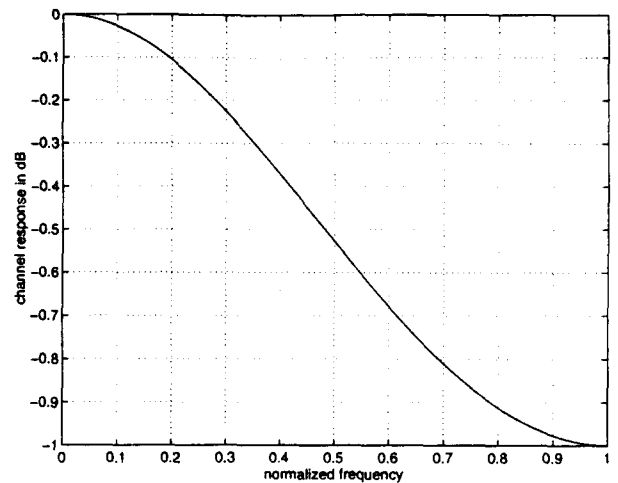


Figure 1: Frequency response of the channel to be equalized

gorithms used 32 bit precision. However, for the modified log-log LMS only 5 bit adders (corresponding to 32 bit dynamic range) are required to compute the gradient. It can be seen from this figure that the performances of the log-log LMS and modified log-log LMS algorithms are close to that of the regular LMS algorithm. The modified log-log LMS algorithm degrades the performance only slightly. In comparison the sign-error LMS, the sign-data LMS and the sign-sign LMS algorithms have a poor performance both in convergence and final MSE. Lowering the step-size μ lowers the final MSE of the sign LMS algorithms but at the same time decreases the convergence speed.

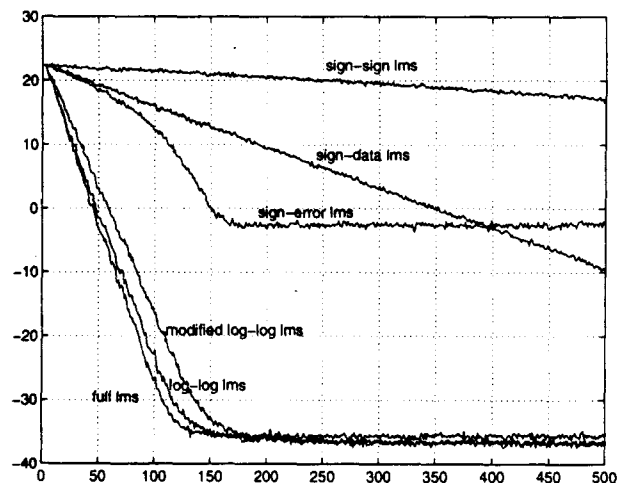


Figure 2: Learning curves ($N = 10$, $\mu = 2^{-11}$, 32 bit precision, input SNR = 40dB)

4. CONCLUSIONS

In this paper we presented a new low complexity log-log LMS algorithm for updating the coefficients of an adaptive filter. The algorithm is similar to the sign based LMS algorithms in that only the update section is modified. However, it results in improvement in both convergence and MSE performance over the sign-based algorithms. In particular, its

performance is close to that of the regular LMS algorithm. A simple modification of this algorithm results in further savings in interconnect routing area while maintaining similar performance. The closeness in the performance of the log-log LMS algorithm and its modification, to that of the regular LMS may be partly explained by the fact that the proposed algorithms try and preserve the dynamic range information of both the error and the data. Besides having a low complexity, the proposed algorithms also require low silicon area to implement.

5. REFERENCES

- [1] K. Laudel, "Performance results of a Low Cost Alternative Equalizer Architecture for 64/256 QAM Demodulation in a CATV Receiver," *NCTA 95 Technical Papers*, 1995.
- [2] A. Gersho, "Adaptive Filtering with Binary Reinforcement," *IEEE Trans. on Info. Thy.*, vol. IT-30, pp. 191-199, March 1984.
- [3] J. R. Trierchler, et. al., *Theory and Design of Adaptive Filters*, John Wiley and Sons, 1987.
- [4] S. Dasgupta and C.R. Johnson, Jr., "Some Comments on the Behavior of Sign-Sign Adaptive Identifiers," *Systems and Control Letters*, vol. 7, pp. 75-82, April 1986.