Finite Precision Effect of Adaptive Algorithm Sigmoidal

José R.S. Fonseca*, André C. B. Cavalcante*, Ewaldo E. C. Santana*, Freire C. S. Raimundo, Allan K. Barros*

Federal University of Maranhão – UFMA
São Luís, Maranhão, Brasil
E-mail: ribamar.sz@hotmail.com

Abstract - Adaptive filtering is currently an important tool in statistical signal processing, especially when it is necessary to process signals from environments with unknown statistics that vary over time. The adaptive filtering study was boosted with the development of the Least Mean Square algorithm (LMS) in 1960. Since then other adaptive algorithms have emerged with superior performance to the LMS algorithm in relation to mismatch and convergence rate. Among them, the algorithm Sigmoidal (SA) that presented superior to the LMS, in relation to the convergence rate and the mismatch in its implementations in infinite precision. In hardware devices, adaptive algorithms are implemented in finite precision, usually with fixed-point arithmetic. When adaptive filters are implemented in finite precision some effects may affect their performance. Ultimately, lead to divergence due to the quantization errors specified in the process of approximation of the values of the variables involved in the adaptive processing of their original values. This article proposes to demonstrate and discuss the performance of the Sigmoidal Adaptive Algorithm (SA) in finite precision, when implemented using fixed-point arithmetic comparing in different word lengths (number of bits): 8 bits, 16 bits and 32 bits.

Keywords - Adaptive Filters, Quantization, Fixed-point Arithmetic, Algorithm Sigmoidal (SA).

I. INTRODUCTION

Many of the problems related to signal processing is solvable through adaptive filtering [10]. An adaptive filter consists of self-adjusting coefficients that satisfy some predetermined criterion after an adaptive process [3]. Typical applications of adaptive filters are: prediction, modeling, noise cancellation, system identification, among others. Two measures stand out in performance analysis of adaptive algorithms: the misadjustment and the convergence rate [2]. Among the adaptive algorithms LMS (Least Mean Square) is a reference and is by far the best known [6]. New adaptive algorithms have emerged with a higher performance than LMS against misadjustment and convergence rate, among these, there is the RLS algorithm (Recursive Least Square) [6]. Another algorithm, which is based on this work is the Sigmoidal (SA), which was shown in [12] its superiority to the LMS comparing the misadjustment and the convergence rate.

The signal processing can be done in analog form (infinite precision) or digital (finite precision). In the analog signal processing of the physical quantities vary along a continuous range of values. In the digital signal processing these amounts assume only discrete values. There are several advantages to digital signal processing compared to the analog signal processing, and main: the robustness and versatility [6]. There are several hardware devices (electronic) using digital signal processing, such as DSPs (digital signal processing), microcontrollers and FPGAs (field programmable gate arrays). One of the major disadvantages of digital technique is that most physical greatness are analogical nature [6].

When working with signals or digital systems carried out an approximation of an analog greatness by a quantized value (discrete), and result in a quantization error [3]. In practical terms, the adaptive algorithms are implemented in finite precision, and often with fixed-point arithmetic [5]. The performance penalty of some adaptive algorithms incurred with the result of the implementation in finite precision has been analyzed [2]. Their behavior can get results or undesirable effects due to quantization errors introduced in the calculations involved in the adaptive processing of these filters [6].

Quantization errors can degrade the performance of an adaptive filter in several ways. For example, they can affect the stability of the filter and ultimately lead to their divergence. They can also degrade the performance of steady-state filter making the MSE to be higher than expected from an infinite precision analysis. The performance degradation tends to be more serious for the RLS algorithm as opposed to the LMS algorithm. This is because for LMS-type filters, noise gradient errors are more significant than the finite precision errors during the transitional phase. However, at steady state, a significant excess squared error can be formed. The transient performance of the RLS algorithm, on the other hand, is more sensitive to finite precision errors and the algorithm may differ as a result [3]. Thus we propose to analyze the effects of finite precision in the SA algorithm. Among the results found in this article we highlight the MSE and the convergence rate of the SA algorithm in finite precision, and for different lengths of words.

This article is organized as follows: In II initially presente the Sigmoidal algorithm and some effects of finite precision adaptive filtering. In III we present the methodology used in the analysis algorithm in finite precision with fixed-point arithmetic. In IV show the main results for the algorithm implemented in different lengths of words. And in V we conducted a brief discussion of the results. VI held the conclusion of the article.
II. BACKGROUND

A. Algorithm Adaptive Sigmoidal (Sa)

The general configuration of an adaptive filtering setting is shown in Figure (1), where $k$ represents the iteration number, $x(k)$ denotes the input signal, $y(k)$ is the output signal of the adaptive filter and $d(k)$ defines the desired signal. The error signal $e(k)$ is calculated $d(k)-y(k)$. The error signal is then used to form a performance function (or goal) that is required by the adaptation algorithm to determine the appropriate update of the filter coefficients $w(k)$ [3].

From this function, $\ln(\cosh \alpha e)$, can generate a family of functions, multiplying the argument and $e$ a positive integer $\alpha$, or when increasing $\alpha$, increases the slope of the performance surface, where the curves of these functions have greater inclination than the curve of a quadratic function of the Least Mean Square algorithm (LMS) [6]. SA to develop the algorithm, we estimate the gradient of $F_k(e)$. Where we get the equation (1):

$$\bar{V}F_k(e) = - \alpha \tanh(\alpha e) x(k).$$

Soon the adaptive algorithm is given by:

$$w(k + 1) = w(k) + \mu \tanh(\alpha e) x(k).$$

Equation (2) is Sigmoidal algorithm. Where $\mu$ is a constant that regulates the speed and stability of the adaptation [6].

B. Effects Of Accuracy Finite (Point Arithmetic Fixed)

There are some differences between the analog form (infinite precision) of the variables and the corresponding value in the digital form (finite precision) of the adaptive filter. All scalar and vector elements in the algorithm AS will divert their correct or unique values, due to the effects of quantization. The quantized error generated at any step in the algorithm is considered a random variable with zero mean, which is independent of any other errors and/or quantities related to the adaptive algorithm. Variations of these errors depend on the type of quantization and arithmetic that will be used in the implementation of the algorithm [6].

Whereas Figure (3), and the structure of the SA algorithm, the equations defining the updating of adaptive filter coefficients of finite precision, is: Equation (3), (4), (5) and (6).

$$d(k) = d(k) - dk,$$  
$$y(k) = \left[w(k)x(k)\right]Q,$$  
$$e(k) = \left[d(k) - y(k)\right]Q,$$  
$$w(k + 1) = \left[w(k) + \mu \tanh(\alpha e)x(k)\right]Q.$$  

Note that $Q$ is the quantization operation, Figure (3). We define the quantization errors or noise in variable quantities and related SA algorithm as follows: Equation (7), (8), (9) and (10).

$$n_d(k) = d(k) - d(k)Q,$$  
$$n_y(k) = y(k) - y(k)Q,$$  
$$n_e(k) = e(k) - e(k)Q,$$  
$$n_w(k) = w(k) - w(k)Q.$$  

Where $n_x$ indicates the noise due to quantization and $\xi$ is the variable or amount related to the algorithm. Assuming that the input signal $x(k)$ does not undergo quantization.
Figure 3: Identification system implemented in finite precision. Where \( x(k) \) denotes the input signal. Blocks \( Q_d \) and \( Q_c \) are quantization of \( b \) bit(s). Thus the quantizer, \( Q_d \), uses \( bd \) bits input data and \( Q_c \) quantizes the coefficients \( b \) bits. The output signal \( y(k) \) indicates the quantized form. The desired signal defined by \( d(k) \) of the unknown system and \( d(k) \) a quantized form. The measurement noise \( nk \) and the error signal quantized \( e(k) \). The coefficients \( w(k) \) of the unknown system and \( w(k) \) adaptive filter [5].

In computers, numbers (real or complex value, integer or fractional values) are represented using binary digits (bits) which takes the value of 0 or 1 [8]. A kind of arithmetic notation used by hardware devices is the fixed-point arithmetic. It determines the number of bits allocated for storing the integer part and fractional part, whereas \((b + 1)\) bits. Figure (4) [8].

Figure (5) shows a model for schematic representation of quantization levels for the case of \( b = 2 \) bits. We consider a value \( L \) threshold. Using \( b \) bits, the range of \([-L, L]\) can be divided into \(2^b\) levels of width \( \Delta = L/2^b \) each. If, moreover, we use one (1) bit signal, the resulting \((b + 1)\) bits divides the interval \([-L, L]\) in \(2^b\) levels. Here we assume rounding operation. [11]

Using fixed-point arithmetic error can be modeled in stochastic process with zero mean with variance given by [3].

III. METHODOLOGY

The implementation process of the adaptive filter SA Fixed Point based on the following flowchart:

1. Step: Before implementing an adaptive filter in finite precision fixed point, you may need to create an adaptive filter floating point with results similar to the adaptive filter in analog form (infinite precision) as reference filter. Important for mounting a scanning configuration. Then come a few steps:

2. Step: Simulate (floating point) the reference filter. In this step determines the appropriate length of the filter step...
size, the variation coefficient of the reference and other filter characteristics.

3. Step: Develop tool to convert the adaptive filter floating point to fixed point.

4. Step: Simulate is the adaptive filter fixed point. At this stage determines the appropriate word length (number of bits) for integer part and fractional part to the set of variables or parameters of the adaptive filter fixed-point.

5. Step: Generate is the code for the adaptive filter fixed point.

6. Step: Validate and verify the project after implementation. Checking the expected overall result.

To use the \( \text{tanh}(e) \) in finite precision using fixed-point arithmetic, it was necessary to realize an approximation of this Taylor series function. Given an indefinitely differentiable function at a certain point in your domain (ie, it exists and is finite the derivative of any order of at \( x = a, f^{(n)}(a) \)) you can always write your Taylor series on the \( a \).

\[
\text{tanh}(e) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)(x-a)^n}{n!} (12).
\]

We apply the adaptive algorithms in an unknown system identification problem, Figure (7) where we have a plant represented by the transfer function polynomial \( P(z) = 0.2037z^{-1} + 0.5926z^{-2} + 0.2037z^{-3} \), whose output is corrupted by noise \( n_k \). Our goal is to find, adaptively, a plant model, \( P(z) \) in finite precision using fixed-point arithmetic. To this end, the SA algorithm used in different word lengths. The input signal is simulated as a random signal uniformly distributed, limited in the range \([-1; 1]\). As used a noise signal order \( 10^{-3} \) with a uniform probability distribution. The desired signal was set as the sum of more noise input signal. The step size parameter for the SA algorithm is \( \mu_{SA} = 0.01 \) and used for the parameter to the value of \( 3 \). In the variable bit here we use a sign bit resulting \( b + 1 \). We made 100 Monte Carlo simulations. For each algorithm in different word lengths (8 bit, 16 bit, 32 bit).

The error in the updating of the integer coefficients of the adaptive algorithm SA quantized can be given by:

\[
w(k + 1)_q = w(k)_q + \mu \text{tanh}(ae)q x(k)_q - n_w(k) (7)\]

Where \( n_w(k) \) is the quantization noise coefficient vector \( w(k) \) in SA algorithm. Here we consider the quantization noise generated at any stage in the algorithm as a random variable with mean zero, which is statistically independent of other errors and/or related amounts in the SA algorithm.

Figure (8) compares the \( \text{tanh} \) function \( \text{tanh}(e) \) and its approximation in series of taylor.

Figure (9), we compare the learning curves, using the finite precision tool developed by fixed-point arithmetic on the learning curve in infinite precision (Floating point).
V. DISCUSSION

These results suggest a supplement of an additional quantization noise due to an increased convergence speed for a specific word length, depending on the problem involved, such feature can also be found in other adaptive algorithms like the LMS. In our problem the misfit level is increased when we implemented the SA algorithm in 8-bits. For 16-bit and 32-bit observed a slight deceleration of speed in relation to the SA into 8-bit. As shown, some challenges on the effects of adaptive filters implemented in finite precision may arise and have harmful effect on performance.

VI. CONCLUSIONS

We analyze the SA algorithm in finite precision using fixed point arithmetic. For this we derive approximation expressions in Taylor series which made it possible to carry out this implementation. The preliminary results found suggest the possible implementation in hardware devices of this algorithm and shows some effects for certain lengths of words. This implies a new practical alternative for solving problems of signal processing found in several areas, such as medicine, military and others. The next steps of this research will be to perform a mathematical analytical exploration of this algorithm in finite precision. As future research we suggest the implementation of this algorithm in hardware device.

ACKNOWLEDGEMENT

To the PPGEE-UFMA(Post-Graduate Program in Electrical Engineering-Federal University of Maranhao), the CNPQ (National Council for Scientific and Technological Development), the FAPEMA (Foundation for Scientific Research and Development of Maranhao) and the PIB (Biological Information Laboratory).

REFERÊNCIAS


