Finite Precision Effect of Adaptive Algorithm Sigmoidal

Authors:
José de Ribamar Silva Fonseca;
Ewaldo E. C. Santana;
André C. B. Cavalcante;
Raimundo C. S. Freire;
Allan K. Barros

Laboratory of Biological Information
- PIB
Federal University of Maranhao –
UFMA
Sao Luís, Maranhão, Brasil
Dear

Due to lack of financial resources in my University, unfortunately it will be impossible to me or any other author to be present in the congress.

Best regards,

José de Ribamar Silva Fonseca.

Teacher advisor:
Allan Kardec Duailibe Barros Filho
Ph.D., Engenharia Elétrica, Universidade de Nagoya, Japão
E-mail: akduailibe@gmail.com
E-mail: cpgee@dee.ufma.br
http://www.ppgee.ufma.br/docente/
Topics

1. INTRODUCTION
2. BACKGROUND
3. METHODOLOGY
4. RESULTS
5. DISCUSSION AND CONCLUSIONS
6. REFERENCE
Representation of the Signals in Finite Precision

The signals can be represented in infinite precision and finite precision.

**Representation Infinite Accuracy:** The signal takes infinite values. The error is null.

**Representation Finite accuracy:** The signal assumes finite values. The error is nonzero.

Figure 1: Signal Representation (Source CBPF–Curso Processamento de Sinais)

Typical examples of signals: Electrocardiogram, Electroencephalogram, Seismic Signals, Voice Sign and Images
Quantizing Error

Quantization is the process of assigning discrete values to a signal whose amplitude varies between infinite values.

Scanning is the process of approximation of an analogue quantity by a quantized value resulting in a quantization error.

Each value of the signal is replaced by the closest level. The more levels the smaller the quantization error will be.

Figure 2: Signal Representation (Source: CBPF—Curso Processamento de Sinais)
Motivation

Digital Hardware Devices work with Representation in Finite Precision.

Examples:

- Digital Signal Processors (DSP’s)
- Programmable Field Gate Arrays (FPGA’s).

Figure 3: DSP e FPGA [Source: CBPF–Curso Processamento de Sinais]
Signal Processing Using Adaptive Filtering

Many signal processing problems are solved through adaptive filtering.

Typical applications: prediction, modeling, noise cancellation, system identification.

Examples of Adaptive Algorithms:
- LMS (Least mean squares);
- RLS (recursive least squares);
- SA (Sigmoidal) e etc.

\[ e(k) = d(k) - y(k) \]  \hspace{1cm} (1)
\[ F_k(e) = \text{Cost Function} \]  \hspace{1cm} (2)
\[ w(k + 1) = w(k) - \mu \nabla F_k \]  \hspace{1cm} (3)

Figure 4: General Configuration of Adaptive Filtering.
Effects of finite precision on adaptive filtering

The errors in the internal calculations due to quantization related to the adaptive algorithm are given by:

\[ n_e(k) = e(k) - e(k)_Q(4) \]
\[ n_w(k) = w(k) - w(k)_Q(5) \]
\[ n_y(k) = y(k) - y(k)_Q(6) \]

Where the subscript Q indicates the quantised form of the given value or vector.

It is assumed that the input signal and the desired signal are not quantized.
The Sigmoidal algorithm (SA).

Cost function \( F_k(e) = \ln(\cosh \alpha e) \).

- We estimate the gradient of \( F_k \)

\[
\hat{\nabla} F_k(e) = -\alpha \tanh(\alpha e) x(k)
\]  
(7)

- Adaptive algorithm given by:

\[
w(k+1) = w(k) - \mu \hat{\nabla} F_k(e)
\]
\[
w(k+1) = w(k) + \mu \alpha \tanh(\alpha e) x(k)
\]  
(8)

- This is the Sigmoidal algorithm.

Figure 6: Porção da superfície tridimensional gerada pela função \( \ln(\cosh \alpha e) \), juntamente com alguns contornos. figura Retirado de: [7]

Methodology

In computers, numbers (real values or complex values, integers or fractions) are represented using binary digits (bits), which take the value of a 0 or 1. One way to represent these values is through fixed-point aritmetics.

![Diagram of Fixed Point Arithmetic]

Figure 7: Fixed point arithmetic
Methodology

✓ Computational Software.
✓ Fixed-point arithmetic; Assuming there is \((b + 1)\) bits
✓ Rounding operation.
✓ The maximum error of quantization will be given in the form:

\[ \Delta = 2^{-b} \]  \hspace{1cm} (14)

✓ The error after each product can be modeled in stochastic process with zero mean, with variance given by:

\[ \sigma^2 = \frac{2^{-2b}}{12} \]  \hspace{1cm} (15)

✓ To use the function \([\tanh(\varepsilon)]\) we approximate the function in Taylor series in fifth degree.

\[ F(k) = \sum_{n=0}^{n} \frac{f^{(n)}(a) \times (k - a)^n}{n!} \]  \hspace{1cm} (16)
Methodology

Application of Adaptive Filtering: System Identification

Input signal \( x(k) \): Uniformly distributed random signal, [-1,1]
Noise \( n_k \): Random Sign Uniformly Distributed Order \( 10^{-3} \).
Adaptation Step size: \( \mu_{\text{Lms}} = 0.3e^{-1} \); \( \mu_{\text{Sa}} = 0.0100 \) \( (\alpha = 3) \)
System Floor: \( P(z) = 0.2037z^{-1} + 0.5926z^{-2} + 0.2037z^{-3}, \)

Figure 8: Adaptive modeling block diagram of a plant.
Methodology

Project SA finite precision

1. Algorithm (Floating point 64-bit)
2. Simulate (Floating point 64-bit)
3. Designing the tool to convert to fixed point
4. Simulate in finite precision (fixed point)
5. Generate code for implementation
6. Validate and verify projects after implementation

Figura 9: Project Workflow
Results

Taylor series approximation in fifth degree of function Sigmoidal SA [tanh(ε)].

Figura 10: Function Comparison Tanh(ε) and Function Tanh(ε) In Taylor series(5º degree)
Results

Learning Curve using the tool developed in Fixed Point on the learning curve in infinite precision.

Figure 11: (SA in precision Infinita and SA Finite precision in fixed point).
Results

Learning curve LMS and SA Algorithm in Finite Precision in Fixed Point Arithmetic (32 bits).

Figure 12: Learning curve SA and LMS at fixed point
Results

In this step we compare the SA in different binary word length in fixed point arithmetic: 8 bits, 16 bits and 32 bits.

Figura 13: SA in different binary word length in fixed-point arithmetic: 8 bits, 16 bits and 32 bits.
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.

<table>
<thead>
<tr>
<th>N de Bits $b_c$</th>
<th>Misajustment</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>2,6978</td>
<td>$2,6978 \times 10^{-6}$</td>
</tr>
<tr>
<td>16</td>
<td>0,0070</td>
<td>$7,0851 \times 10^{-9}$</td>
</tr>
<tr>
<td>32</td>
<td>0,0063</td>
<td>$6,3681 \times 10^{-9}$</td>
</tr>
</tbody>
</table>
Conclusion

• 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.
References


References


Thank you.