A Modified Affine Arithmetic Method for Computational Error Analysis

Azar Heidari, Arash Ahmadi, Shahpour Alirezaee* and Majid Ahmadi
Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, Canada
*Corresponding author

Keywords: Computational error, Interval arithmetic, Affine arithmetic.

Abstract. Computational precision always has been a concern in digital systems design and optimization. This study presents a modified Affine Arithmetic method to calculate uncertainty range in the output of arithmetic units. We have considered five case studies, namely: quadratic equation, RGB to CrYCb system, fourth-order equation, multivariate polynomial functions and low pass filter, by which the proposed method is evaluated. Presented results indicate the modified affine arithmetic can outperform the traditional methods.

Introduction

Digital computation is considered to have a limited precision. Most of DSP algorithms are preliminary developed in floating-point arithmetic then implemented on low cost fixed-point platforms. To optimize the word length in a design, one should compute the range of the uncertainty in the output. It is often impossible or unfeasible to predict mathematically the exact magnitude of the round off or truncation errors transported to the output of a numerical task. K. Kum et.al [1] have developed a word length optimization and high-level synthesis algorithm to minimize the hardware implementation cost and reduce the optimization time. Authors in [2] present an algorithm to optimize the bit widths of fixed-point variables for low power system design in a system level ASIC design environment. In [3] range arithmetic is utilized to calculate signal intervals. Generally, these approaches can provide more accurate solutions and are faster compared with the methods presented in [1] and [2]. Therefore, analytical optimization methods are more attractive for large scale designs. One approach to deal with such numerical errors is a technique called interval arithmetic (IA) [4]. Although it is an effective and simple method, it does not consider dependency between error sources, which might result in error overestimation. In [5] Fang et.al proposed a method in which probability density function of the output round off noise is estimated using central limit theorem.

The rest of this paper is organized as follows. Section 2 introduces the necessary background of the affine arithmetic (AA) method and quantized affine arithmetic (QAA). We describe AA and QAA for functions in detail. The proposed approach is presented in Section 3, where a simple method to approximate the filters’ output in presence of computational errors is discussed in Section 4. For evaluating the proposed approach we have considered five case studies in Section 5 and finally paper is concluded in Section 6.

Interval Methods

Assuming that the variation ranges of the input variables are known, error bound analysis approaches aim to predict the variation range of the output data. Instead of a single value, every number is represented by a variation interval between the upper and lower bounds. These methods are concerned with data range dilation and contraction by data propagation through computation tree of the system with no information about where the actual value might be placed in the range.
Interval Arithmetic

Interval Arithmetic (IA), was originally introduced in [6], as a tool for automatic error control arisen from rounding and truncation. In IA, every real value, \( x \), is represented by an interval \( \hat{x} = [x_l, x_h] \), where \( x_l < x < x_h \). Accordingly, arithmetic operations are performed on these intervals such that each result interval is guaranteed to contain the value of the actual quantity. Unfortunately, computed error bounds can be overestimated by IA in practice, because dependencies of the data values are not considered. In other words, since there is no trace of the error sources in the intervals, error sources which originate from parallel branches in the computation tree, might share some common components that are not considered in the interval computation. As a trivial example, consider \( x - x = 0 \), which holds for each \( x \in [-1, 1] \), but \( \hat{x} - \hat{x} \) for \( \hat{x} \in [-1, 1] \) results in \([-2, +2]\) in the IA method. Another source of overestimation is called the wrapping effect. This effect appears when intermediate results of a computation are enclosed in intervals. These types of over-estimations are classified as one of major problems in the IA treatment of differential equations.

Affine Arithmetic

Affine Arithmetic is an extension of the traditional IA. In each operation, AA keeps track of the source and signal amplitude of all the uncertainties that affect each variable. A given affine value, \( \hat{x} \), is mathematically expressed as:

\[
\hat{x} = x_0 + \sum_{i=1}^{n} x_i \varepsilon_i \quad -1 \leq \varepsilon_i \leq 1
\]  

(1)

where \( x_0 \) is the central value, \( n \) is the number of noise terms and \( \varepsilon_i \) and \( x_i \) are the identifier and amplitude of the \( i \)th noise term. In (1) each identifier represents an independent uncertainty contained in the interval \([-1, 1]\). For example, if the input signal is bounded by \([-2, 1]\), it is presented in AA as:

\[
\hat{x} = -0.5 + 1.5\varepsilon \quad -1 \leq \varepsilon \leq 1
\]  

(2)

The main advantage of AA is that it alleviates the so-called dependency problem of IA. For example reconsider \( x = [-1, 1] \) and \( y = x - x \). As we saw before, the IA results \( y = [-2, 2] \) but the AA results as follows:

\[
\hat{x} = 0 + \varepsilon \quad -1 \leq \varepsilon \leq 1 \quad y = \hat{x} - \hat{x} = \varepsilon - \varepsilon = 0.
\]  

(3)

Although the AA can resolve the dependency problem it has a crucial issue which is explained by an example. Suppose \( x = 2 + \varepsilon_1 \), \( y = 1 + 2 \varepsilon_2 \) and the output is \( z = x \times y \). i.e : 

\[
\hat{z} = (2 + \varepsilon_1)(1 + 2 \varepsilon_2) = 2 + \varepsilon_1 + 4 \varepsilon_2 + 2 \varepsilon_1 \varepsilon_2.
\]  

(4)

As can be seen, although the approach is based on linear relationships, the fourth term is nonlinear. Therefore this nonlinear element must be replaced by a new symbol as follows:

\[
\hat{z} = 2 + \varepsilon_1 + 4 \varepsilon_2 + 2 \varepsilon_3.
\]  

(5)

If \( \hat{z} \) multiplies in other signal with a different symbol of noise, the new result has 6 noise symbols and if it continues this way, the final result will contain a large number of noise symbols. Similar problem arises for other nonlinear relationships; accordingly, modifications are presented for AA method to resolve its basic limitations.
Quantized Affine Arithmetic

Quantized affine arithmetic (QAA) is a straightforward modification of the AA. This method keeps main symbol of the input signal errors and prevents new symbols to be created. For example, if \( \hat{x} = x + x\varepsilon_1 \), \( \hat{y} = y + y\varepsilon_1 \) and the output relationship \( z = x \times y \) the result of the AA is as follows:

\[
\hat{z} = x_0 y_0 + x_1 y_0 \varepsilon_1 + x_0 y_1 \varepsilon_2 + x_1 y_1 \varepsilon_3.
\]  

(6)

in which AA replaced the multiplication between symbols by a new symbols (here is \( \varepsilon_3 \)). The similar output is represented in QAA as:

\[
\hat{z} = Q_0^R (x_0 y_0 + x_1 y_0 \varepsilon_1 + x_0 y_1 \varepsilon_2).
\]  

(7)

where \( Q_0^R \) represents quantization function. As a matter of fact, this method converts the coefficients of the symbols \( \varepsilon_i \) (here \( \varepsilon_1, \varepsilon_2 \)), to integer numbers and omits the new symbols. Therefore next equation does not have any new symbols and quantizes the coefficients to integer numbers (\( \varepsilon_3 \)). However there is a main problem in this method. If we assume \( m \) and \( n \) are integer numbers and \( n > m \), the numbers from \( m \) to \( m/2 \) are equal to \( m \) and the numbers from \( m/2 \) to \( n \) are equal with \( n \), and as a result, estimation of the exact results in nonlinear equations is impossible. For instance:

Affine Arithmetic: \( \hat{x} = 0 + \varepsilon_1 \), \( \hat{y} = 1.6 + 1.6\varepsilon_2 \), \( z = xy \).  

(8)

Quantized Affine Arithmetic: \( \hat{z} = 1.6\varepsilon_1 + 1.6\varepsilon_3 \), \( \hat{z} = [-3.2, 3.2] \).  

(9)

The length of the interval in QAA is less than AA and it does not include all results. As a result, although QAA is a simple and fast approach, it is not always accurate.

The Proposed Approach

Major problem of AA is overestimation in nonlinear equations. Every variable contains a noise symbol (\( \varepsilon \)) in equations and when \( \varepsilon_i \) is multiplied by \( \varepsilon_j \) produces a new symbol (\( \varepsilon_{\text{new}} \)). The proposed method utilizes AA method in multivariable equations where a lower and upper bound of the input ranges in combination with AA method. Although lower and upper bounds change the minimum and maximum of the equations, the variable with the highest exponent has the strongest effect on the output range of the function. We name it the “main variable” but if all variables have the same effect, the variable with largest length of interval is called the main variable. The first step in this method is to find the input-output relationship. Unlike other methods, this method does not need to divide the equations into primitive or smaller parts to compute each part separately. Instead, a straight relationship between input and output is adequate.

Specify the main variable is the next step. Except for the main variable, all the combinations which can be made by endpoints of the variables intervals; must be considered. Similar to IA, this method considers only two endpoints of the intervals, instead of all values in the intervals. For instance, for \( a \in [a_1, a_2] \) and \( b \in [b_1, b_2] \) intervals, all combinations are, \( a_1 b_1, a_1 b_2, a_2 b_1, a_2 b_2 \).

For an input-output relationship with \( n \) variables, there are \( 2^n - 1 \) equations. These equations are functions of the main variable. We should change the main variable like equation (1). Put main variable in each function and calculate lower and upper bounds of each equation. The final interval is identified by comparing all intervals.

In addition, we have:

\[
a + a = 2a_0 + 2 \sum_{k=1}^{n} a_k \varepsilon_k.
\]  

(10)

As we can see the number of noise symbols is half of AA method when the variables are different. In subtraction result is:
\[ a - a = (a_0 - a_0) + \left( \sum_{k=1}^{n_a} a_k \cdot \varepsilon_k - \sum_{k=1}^{n_a} a_k \cdot \varepsilon_k \right) = 0. \]  

(11)

Multiplication is a challenging operation in AA method, because it produces extra noise symbols which can result in an overestimation, as:

\[ a \cdot a = (a_0 \cdot a_0) + 2 \sum_{k=1}^{n_a} a_0 \cdot a_k \cdot \varepsilon_k + \left( \sum_{k=1}^{n_a} a_k \right)^2 \cdot \varepsilon_{\text{max}}. \]  

(12)

This method is applicable for affine and non-affine equations, where in all other methods the non-affine equations are problematic. Despite its capability of working with non-affine computation, in the case of recursive equations or feedback systems, other methods are slow. Although other methods can be used for feedback systems and we can get acceptable results, but the speed of output interval computation is low.

Case Studies and Numerical Results

For evaluating the proposed approach, we have considered five case studies including: Quadratic equation, RGB to CrYCb system, Fourth-order equation, Multivariate polynomial functions, and Low pass filter.

Quadratic Equation

Consider the following quadratic equation with the input and coefficient error bounds as:

\[ y = ax^2 + bx + c \quad x \in [2, 4], \quad a \in [9,10], \quad b \in [-6,-4], \quad c \in [6,7]. \]  

(13)

The y output ranges can be computed using the mentioned methods. The results indicate the output range as [18,159] for IA, [3,151] for QAA, [3,151] for AA, and [12,151] for the proposed approach where the best results are achieved by the proposed method.

RGB to CrYCb System

As another case of study, we consider is a linear system that converts R, G, B signals to Cr, Y, Cb signals as the following transformations:

\[ C_r = 0.5R - 0.4542G - 0.0458B \quad Y = 0.222R + 0.7067G + 0.0713B \quad C_b = -0.119R - 0.38G + 0.5B \]

Suppose the following input ranges for R, G, and B:

\[ R = [63.5,77.5] \quad G = [58,80] \quad B = [71,89] \]

We aim to find the output ranges for Cr, Y, and C. All combinations of the variables are: \( \{R,B,R,B,R,B,R,B\} \)

Therefore, there are 4 conditions for each output and consequently 12 equations of G, for Cr and Cb, Y. The results are presented in Table1.

<table>
<thead>
<tr>
<th>Output</th>
<th>IA</th>
<th>QAA</th>
<th>AA</th>
<th>Proposed approach</th>
<th>Exact value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>[60.1479,80.0867]</td>
<td>[60.82]</td>
<td>[60.148,80.0866]</td>
<td>[60.1479,80.0867]</td>
<td>[60.1969,80.0866]</td>
</tr>
</tbody>
</table>

Table 1. The results of RGB to CrYCb converter.

Fourth-Order Equation. In this section, three multivariate polynomial functions are considered. First equation is an image rejection unit as follows:

\[ f_i(x) = 16384(X_i^4 + X_j^4) + 64767(X_i^2 - X_j^2) + X_i - X_j + 57344X_iX_j(X_i - X_j). \quad X = [0,1]^2 \]

Second equation is Matyas function:

\[ f_m(x) = \frac{1}{2}(x_1^2 + x_2^2) - 0.1 \cos(2\pi x_1) - 0.1 \cos(2\pi x_2). \]
The last equation is Booth function as:

\[ f_4(x) = 0.26(X_1^2 + X_2^2) - 0.48X_1X_2 \quad X = [-100, 100]^2 \]

The results, presented in Table 2, indicate our approach outperforms IA and AA in the range analysis results.

<table>
<thead>
<tr>
<th>Function</th>
<th>IA method</th>
<th>AA method</th>
<th>Proposed method</th>
<th>Accurate range</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_1(x) )</td>
<td>[-122112, 122112]</td>
<td>[-148351.5, 152447.5]</td>
<td>[-5.5e4, 8.1e4]</td>
<td>[-5.51e4, 8.79e4]</td>
</tr>
<tr>
<td>( f_2(x) )</td>
<td>[-10000, 10000]</td>
<td>[-10000, 10000]</td>
<td>[400, 10000]</td>
<td>[0, 10^4]</td>
</tr>
<tr>
<td>( f_3(x) )</td>
<td>[-1726, 2594]</td>
<td>[-2446, 2594]</td>
<td>[-766, 2594]</td>
<td>[0, 2587.9]</td>
</tr>
</tbody>
</table>

**Low pass filter.** This example compares capability of different methods for error analysis in an IIR low pass filter (Figure 1). The coefficients of the sample filter for each section are:

First section: \( b_0 = 1, b_1 = 2, b_2 = 1, a_3 = 1, a_4 = -0.1739a_5 = 0.1116 \)

Second section: \( b_0 = 1, b_1 = 1, b_2 = 0, a_3 = 1, a_4 = -0.0787a_5 = 0 \)

Third section: \( b_0 = 1, b_1 = 2, b_2 = 1, a_3 = 1, a_4 = -0.2397a_5 = 0.532 \)

The error bound results are shown in Table 3 where the results indicate the proposed approach outperforms the others.

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Methods</th>
<th>X(0)=[1.9,4.1]</th>
<th>X(1)=[0.7,3.3]</th>
<th>X(2)=[2.8,5.2]</th>
<th>X(3)=[0.6,3.4]</th>
<th>X(3)=[0.5,2.7]</th>
</tr>
</thead>
<tbody>
<tr>
<td>IA</td>
<td>[1.9,4.1]</td>
<td>[11.135,25.81]</td>
<td>[27.999,73.87]</td>
<td>[37.62,130.9]</td>
<td>[21.147,167.84]</td>
<td></td>
</tr>
<tr>
<td>AA</td>
<td>[1.9,4.1]</td>
<td>[11.135,25.81]</td>
<td>[29.415,72.46]</td>
<td>[47.4,121.23]</td>
<td>[51.96,136.99]</td>
<td></td>
</tr>
<tr>
<td><strong>Proposed approach</strong></td>
<td>[1.9,4.1]</td>
<td>[11.23,25.7]</td>
<td>[28.2,73.55]</td>
<td>[39,129,578]</td>
<td>[24.95,164.567]</td>
<td></td>
</tr>
<tr>
<td><strong>Exact Value</strong></td>
<td>[1.9,4.1]</td>
<td>[11.23,25.82]</td>
<td>[29.64,73.823]</td>
<td>[42.21,130.7]</td>
<td>[28.47,166.398]</td>
<td></td>
</tr>
</tbody>
</table>

Figure 1. Fifth-order IIR Butterworth filter.

**Conclusions**

In this paper presented a new method of interval calculation. The proposed method is a modified version of the AA method for multivariable functions which calculate uncertainty range in the output of arithmetic units. Unlike the AA method, it does not have overestimation problem as much as AA method and the results indicate higher performance in comparison with the AA method for linear and nonlinear systems. We have considered five case studies, including Quadratic equation, RGB to CrYCb system, Fourth-order equation, Multivariate polynomial functions, and Low pass filter. The results indicate the modified affine arithmetic roughly outperforms the traditional methods.

**References**


