Computers and Electrical Engineering 69 (2018) 194–211

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Adaptive segmentation methodology for hardware function evaluators☆

J.M. Trejo-Arellano[a], J. Vázquez Castillo[*,b], O. Longoria-Gandara[a], R. Carrasco-Alvarez[c], C.A. Gutiérrez[d], A. Castillo Atoche[e]

[a] Instituto Tecnológico y de Estudios Superiores de Occidente, Periférico Sur Manuel Gómez Morín 8585, C.P. 45604, Tlaquepaque, Jal., MEXICO
[b] Universidad de Quintana Roo, Boulevard Bahía s/n esq. Ignacio Comonfort, C.P. 77019, Chetumal, Q. Roo., MÉXICO
[c] Universidad de Guadalajara, Blvd. Marcelino García Barragán 1421, C.P. 44430, Guadalajara, Jal., MÉXICO
[d] Universidad Autónoma de San Luis Potosí, Av. Salvador Nava Martinez s/n, C.P. 78290, San Luis Potosí SLP, MÉXICO
[e] Universidad Autónoma de Yucatán, Av. Industrias No Contaminantes s/n, C.P. 97000, Mérida, Yuc., MÉXICO

ARTICLE INFO

Keywords:
Function approximation
Piecewise polynomial
Hardware optimization
Polynomial segmentation
Mathematical functions
Function evaluators
Surrogate modeling
Particle swarm optimization

ABSTRACT

This paper presents a new adaptive function segmentation methodology to evaluate mathematical functions in hardware systems through piece-wise polynomial approximation methods. In contrast to conventional segmentation techniques, this methodology automatically adjusts the segmentation strategy through a function shape analysis based on the first- and second-order derivatives. Additionally, a particle swarm optimization algorithm is implemented to search for the best segmentation parameters that satisfy the designer-given signal-to-quantization-noise ratio specification and minimize the number of polynomials. The main advantages are a significant lookup table size reduction, increased approximation accuracy of highly nonlinear sections, and the automatic generation of a hierarchy-less segmentation solution. Hence, the proposed methodology enables efficient development of hardware accelerated surrogate models such as wireless channel emulators and other signal processing applications on inexpensive platforms that rely on fixed-point number representation as a compromise between performance, and output accuracy.

1. Introduction

Nowadays, digital signal processing algorithms use complex blocks associated with the evaluation of transcendental functions [1–3]. In this sense, modeling of wireless communication channels is accomplished using various methods such as ray tracing [4,5], sum-of-cisoids (complex exponentials) [6,7], and others. For example, the evaluation accuracy of the $\sin(\cdot)$ and $\cos(\cdot)$ functions employed in channel models based on sum-of-cisoids is a primary concern. Additionally, the hardware implementation of Weibull fading channel emulators, which are used for modeling vehicle-to-vehicle (V2V) channels [8], is significantly complex due to the evaluation of $\ln(\cdot)$, $\sqrt{\cdot}$, $1/x$, and $\exp(\cdot)$ functions [9,10]. Likewise, the efficient hardware implementation of algorithms based on algebraic matrix operations such as QR decomposition (QRD) [11,12], is sensitive to the evaluation accuracy of the functions $\sqrt{\cdot}$ and $1/\sqrt{\cdot}$ [13]. Also, function evaluators are implemented to deal with high-complexity blocks of communication systems (e.g. predistorters implemented in [14,15]), and for hardware accelerator blocks employed in general-purpose computing and graphic processor unit (GPU) applications [16].

☆ Reviews processed and recommended for publication to the editor-in-chief by area editor Dr. E. Cabal-Yepez.

* Corresponding author.
E-mail address: jvazquez@uqroo.edu.mx (J. Vázquez Castillo).

https://doi.org/10.1016/j.compeleceng.2018.04.024
Received 15 December 2017; Received in revised form 28 April 2018; Accepted 30 April 2018

Currently, there are several methods for evaluating transcendental functions. Although some of them offer certain advantages, they are also subject to disadvantages that make them unsuitable for applications that require high accuracy and substantial computing throughput. Iterative methods such as CORDIC (COordinate Rotation DIgital Computer) allow for the efficient evaluation of transcendental functions [1,17,18]. However, output accuracy is highly dependent on the number of iterations that the algorithm is executed. This characteristic represents a significant drawback that limits the development of hardware architectures for real-time computing applications. An alternative methodology for evaluating transcendental functions is via look-up tables (LUT) [2,3,10,19]; this is arguably the simplest and easiest way to implement function evaluation blocks; however, the memory size needed for allocating function values increases along with the output accuracy requirement. Recent signal processing algorithms implement function evaluation using different strategies; e.g., [12] implements a function evaluation block using bivariate numerical function evaluation methodology. However, the performance of this methodology is not corroborated via the approximation of transcendental arbitrary functions, which complicate a complete analysis of the proposal. On the other hand, [20] computes arithmetic functions using stochastic logic by series expansion. In this case, arithmetic functions are approximated via truncated Maclaurin series polynomials. Today, the latency of the architectures generated with this methodology is an ongoing discussion, and the re-use of architectures for evaluating different functions is difficult to achieve.

On the other hand, piecewise polynomial approximation (PPA) is an alternative method for evaluating transcendental functions. PPA methods offer flexible design trade-offs between computing speed, area, output accuracy, and hardware architecture utilization because the design of the polynomial evaluator does not change across functions. Approximating functions using PPA methods requires the input evaluation interval to be partitioned into multiple segments. Each of these segments is then approximated using a low-degree polynomial, which is addressed through the hardware polynomial evaluator according to the function input value. In this sense, the accuracy achieved using the PPA approximation methodology significantly depends on the segmentation method employed; i.e., sizable approximation errors might be introduced when an inadequate segmentation strategy is employed, resulting in a reduced signal-to-quantization-noise ratio (SQNR) of the function evaluation block.

Today, the most popular segmentation methodology for PPA is called hierarchical segmentation method (HSM) [21], which combines the more basic segmentation methodologies known as uniform and non-uniform-by-power-of-two. In principle, any function could be segmented out through all these methodologies; however, the downside is that these methods are not sensitive to the function shape, therefore causing substantial accuracy loss and SQNR degradation of the hardware implementation. In this sense, this paper presents a new segmentation methodology for arbitrary transcendental functions, which addresses the segmentation process as a constrained optimization problem where the goal is to determine the minimum number of segments according to design objectives such as SQNR and hardware area. This is achieved through an automated function shape analysis using the first- and second-order derivatives within the specified evaluation interval.

This paper improves on the work and results presented in [22] through the implementation of a global search algorithm to optimize the segmentation strategy and minimize hardware resources. In addition, a cost function is defined in terms of the segmentation design variables to assess the compliance of the segmentation strategy with respect to the SQNR requirements. Furthermore, a broader set of test functions is presented to emphasize the benefits regarding increased accuracy and the reduction of hardware resources achieved through the proposed function segmentation methodology.

The main contributions of this paper are summarized as follows:

- A new adaptive function segmentation methodology (AFSM), for evaluating arbitrary mathematical functions via PPA.
- A shape analysis procedure for arbitrary functions based on first- and second-order derivatives.
- The introduction of an optimization algorithm and a cost function for finding the best parameters of the segmentation algorithm according to specific design objectives for hardware resources optimization.

The simulation results show that the AFSM provides better segmentation performance and higher SQNR with lower hardware resource consumption in comparison to state-of-the-art segmentation methodologies; therefore, the AFSM represents an excellent alternative for implementing high-accuracy PPA-based transcendental function evaluators for sophisticated digital signal processing algorithms implemented on hardware.

This paper is organized as follows: State-of-the-art approximation methods for implementing function evaluators in hardware are described in Section 2. The proposed adaptive function segmentation method for hardware resource optimization is presented in Section 3. Correspondingly, the segmentation and approximation results of the introduced segmentation methodology are presented in Section 4. Finally, conclusions are provided in Section 5.

2. Background

The approximation accuracy of a mathematical function through PPA methods highly depends on the function, system word length, number of segments, and the segmentation methodology employed. The segmentation methodologies most commonly used for hardware-based function evaluators are uniform, non-uniform by powers of two, and the HSM proposed in [21]. Consider a continuous function $f(x)$, with first- and second-order derivatives, where $x \in X$ and $X = [x_L, x_H]$. Uniform segmentation methodology divides the function interval X, into equal-sized segments, whereas non-uniform-by-power-of-two segmentation methodology decreases the size of subsequent segments within X according to a geometric progression with a common ratio of 1/2. In the latter, the segmentation can progress either from x_L to x_H or vice-versa.

In Fig. 1, basic segmentation methodologies show lousy performance when dealing with functions that present non-monotonic