Interval-based analysis and word-length optimization

# of non-linear systems with control-flow structures

***J.A. López, E. Sedano, C. Carreras, and C. López**

Dpto. Ingeniería Electrónica, Universidad Politécnica de Madrid. 28040 Madrid, Spain.

*Corresponding author: juanant@die.upm.es

Abstract

The techniques based on extensions of interval computations allow fast and accurate analysis of the behavior of complex systems. Some of the most recent works in this area have presented procedures to evaluate systems with smooth non-linearities. We take this approach a step further by introducing a methodology that combines Multi-Element Generalized Polynomial Chaos (ME-gPC) and Statistical Modified Affine Arithmetic (MAA). This methodology allows modeling systems with highly non-linear operators and/or control-flow structures. It has been implemented in our modular and automated analysis framework, HOPLITE, so that it can be used to estimate the dynamic range, quantization noise and sensitivity of systems containing the aforementioned control-flow blocks. With this approach we have obtained in case studies with non-linear operators a deviation of only 0.04% with respect to the simulation-based reference values, which proves the accuracy of our approach.

Keywords: Interval Computation, Polynomial Chaos, Affine Arithmetic, Digital Signal Processing, Fixed-Point, Quantization, FPGA Implementation.

Introduction

In an industry where time-to-market is critical, the design and implementation of efficient and reliable Digital Signal Processing (DSP) systems can make the difference between success and failure. In addition, fixed-point computations are preferred when such systems are implemented on FPGAs and ASICs due to the lower implementation cost and power consumption, and higher performance with respect to its floating-point alternative. However, finding a fast and general way for transforming floating-point system descriptions to efficient fixed-point implementations remains an open issue. The analysis and selection of optimized word-lengths is an important and time-consuming step in the design of DSP and VLSI systems. Studies indicate that fixed-point refinement can take up to 25% to 50% of the overall development time [1]. Thus, automating and accelerating this process is strongly desirable.

During the past decades there has been a lot of work on the analytical characterization of the different structures of the DSP subsystems using mathematical expressions [1-20]. These studies provide guidelines to optimize these blocks, but they fail to provide results for the newly-developed (typically complex) structures, as well as for the complete (large) systems. To try to overcome this issue, a number of proposals has recently appeared. They are aimed at developing fast and accurate computation models aimed at providing the optimized word-lengths for the specific system that will be implemented.

Figure 1 outlines the main parts of this Word-Length Optimization (WLO) process [16]. Three major areas are easily identified: (i) Determining the dynamic range of the signals of the system, in order to allocate the integer word-length of each variable; (ii) assigning the number of bits of the fractional word-lengths; and (iii) obtaining the statistical deviation (quantization noise) and determining the validity of the results. None of these three areas is trivial, and each of them is a large field of research on its own.

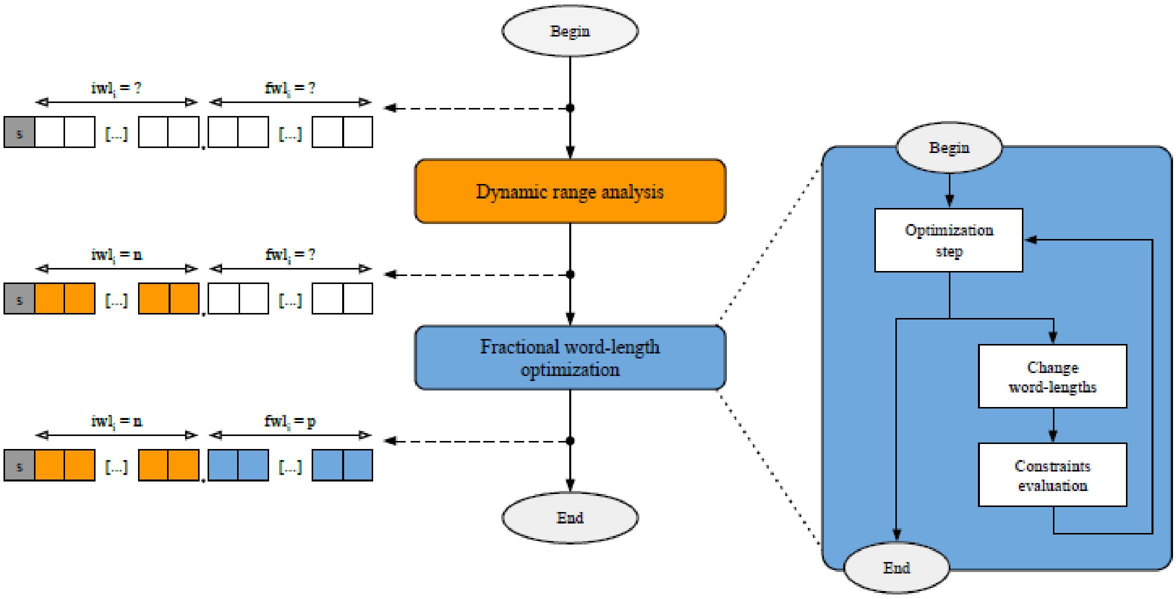


Figure 1. Fixed-point word-length optimization flow

In practice, the WLO process is commonly split in two parts: First, a computational accuracy constraint is determined according to the application performance, and then a WLO technique is applied using this constraint. Such modern WLO techniques are classified in two groups: simulation-based approaches, and analytical (or hybrid) ones.

Simulation-based techniques [2, 3] for modeling the quantization are the most reliable and general approaches, but also the slowest. In order to obtain accurate models, large input data sets are usually required. This makes simulation-based methods impractical for WLO, since estimations must be repeated many times with different combinations of word-lengths as the optimization progresses.

Modern analytical or hybrid techniques are several orders of magnitude faster than the simulation-based ones, but they are limited a given type of systems [4-7]. They perform separate analysis of the word-lengths required for the integer part (to represent the dynamic range of the signals) and the fractional one (to comply with the specified round-off constraint). The integer word-lengths are determined using range propagation or interval arithmetic. The fractional word-lengths are determined using a number of techniques, such as the Perturbation Theory [4], System Transformations [7], Arithmetic Transformations [17], and Handelman Representations [18]. Different Extensions of Interval Computations based on Affine Arithmetic (AA) [5, 6] have also provided very fast and accurate results, but they must be applied according to the characteristics of the system to be evaluated (linear, quasi-linear, polynomial, or strongly non-linear) [16].

The structure of the full version of the paper will be as follows: The models based on extensions of AA used to evaluate the different types of systems will be explained in separate subsections of Section 2. It will also be shown that the non-linear computations need the application of Polynomial Chaos techniques to provide accurate results. Section 3 will explain some of the main applications that can be performed using our AA-based analysis, such as the sensitivity-driven optimization. The tool used for the propagation and computation of the results will be briefly desbribed in Section 4. Two of its main features will be highlighted: its modular implementation and the gradual computation of the results, since they are of particular importance for High Performance Computing (HPC) and the analysis of big data applications. Finally, Section 5 will provide the conclusions and summarizes this work.

Theoretical background on Extensions of Interval Computations

The evaluation of the quantization techniques using Extensions of Interval Computations has been rapidly progressing during the past years, and different new methodologies have been suggested to improve the quality and accuracy of the solutions, as well as to broaden the scope of the systems that can be addressed using them.

The first of such extensions is Affine Arithmetic (AA). AA has been originally suggested for the evaluation and characterization of the linear systems, and has shown to provide among the fastest computation times [10, 11]. However, AA is not able to capture of the correlations of the nonlinear operations. To overcome this fact, Modified Affine Arithmetic (MAA) has been proposed instead [5, 11, 19]. MAA contains higher-order terms that keep track of the results of the non-linear operations. However, these higher-order terms are not orthonormal, so the propagation of the affine terms provides misleading results.

A key feature for the accurate propagation of the higher-order terms is the incorporation of the Polynomial Chaos Expansions (PCE) techniques. The intervals of AA are included in the computation as parameters of the orthonormal polynomials of PCE, thus allowing easy propagation of the coefficients through the nonlinear system [16, 20]. This approach has been applied to dynamic range estimation [20], and to the analysis of the quantization noise for small, sequential systems [16]. However, PCE still fails to efficiently handle systems with discontinuities, and is not capable of modeling control-flow operations. Multi-Element generalized Polynomial Chaos (ME-gPC) is able to produce accurate models for discontinuous systems [16], as will be explained below. In this Section the mathematical background for AA, MAA, PCE and ME-gPC is given.

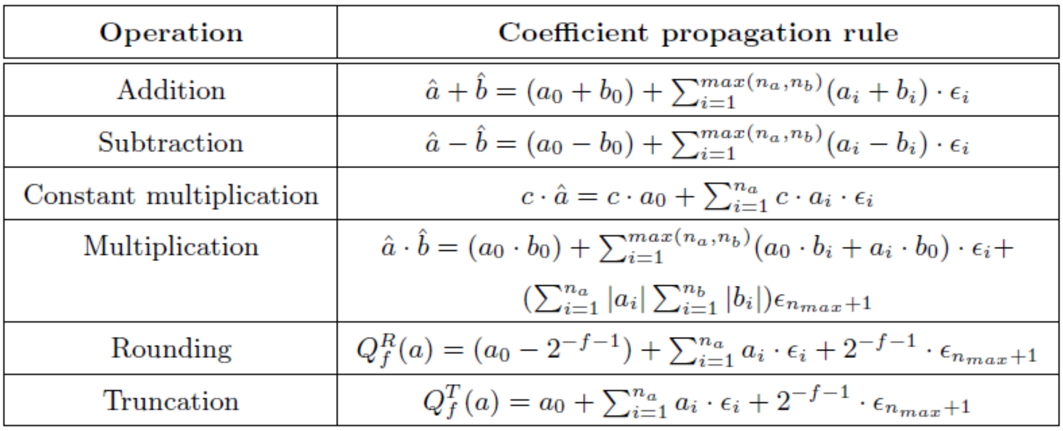
Affine Arithmetic (AA)

An affine form is defined as a polynomial expansion of order one where the independent variables are uniformly distributed in the interval [−1, 1]. Affine arithmetic is capable of capturing the correlation between intervals after affine operations (i.e. linear). A first-order affine form is expressed as [6]:

$$\hat{a} = a_0 + \sum_{i=1}^{n_a} a_i \varepsilon_i \tag{1}$$

The mean value is given by a_0, the terms ε_i are the independent sources of uncertainty and the coefficients a_i are the amplitudes of these uncertainties. The uncertainty sources can represent the variations of the signal or the RON. The basic operations between two affine forms $\hat{a}$ y $\hat{b}$ are summarized in Table 1 [5, 6]. The instructions supported by this methodology are either linear [5, 6] or smooth non-linear [9], meaning that their behavior can be approximated by linear models. The terms n_{max} refer to the maximum number of noise terms present in the affine forms.

Table 1. Coefficient propagation rules of Affine Arithmetic



Operation	Coefficient propagation rule
Addition	$\hat{a}+\hat{b}=(a_0+b_0)+\sum_{i=1}^{max(n_a,n_b)}(a_i+b_i)\cdot\epsilon_i$
Subtraction	$\hat{a}-\hat{b}=(a_0-b_0)+\sum_{i=1}^{max(n_a,n_b)}(a_i-b_i)\cdot\epsilon_i$
Constant multiplication	$c\cdot\hat{a}=c\cdot a_0+\sum_{i=1}^{n_a}c\cdot a_i\cdot\epsilon_i$
Multiplication	$\hat{a}\cdot\hat{b}=(a_0\cdot b_0)+\sum_{i=1}^{max(n_a,n_b)}(a_0\cdot b_i+a_i\cdot b_0)\cdot\epsilon_i+$ $(\sum_{i=1}^{n_a}\lvert a_i\rvert\sum_{i=1}^{n_b}\lvert b_i\rvert)\epsilon_{n_{max}+1}$
Rounding	$Q_f^R(a)=(a_0-2^{-f-1})+\sum_{i=1}^{n_a}a_i\cdot\epsilon_i+2^{-f-1}\cdot\epsilon_{n_{max}+1}$
Truncation	$Q_f^T(a)=a_0+\sum_{i=1}^{n_a}a_i\cdot\epsilon_i+2^{-f-1}\cdot\epsilon_{n_{max}+1}$

Linear operations (addition, subtraction and constant multiplication) are executed in a precise manner. However, after performing the nonlinear operations the temporal correlations of the input signals are lost [5]. The result of executing non-linear operations over uniform distributions is typically non-uniform so it is theoretically impossible to represent it as a linear combination of uniform distributions. In order to alleviate this shortage, MAA [21] introduces higher order polynomials to capture the correlations among the signals.

Modified Affine Arithmetic (MAA)

MAA was initially used for polynomial evaluation and algebraic curve plotting in 2D [21]. Given two affine forms:

$$\hat{a}=a_0+a_1\varepsilon_a\,,\ \hat{b}=b_0+b_1\varepsilon_b \tag{2}$$

ε_a and ε_b are the noise terms bounded in the interval [–1, 1], a_0 and b_0 are the means of both variables and a_1 and b_1 represent the variations of the signals over the mean values. The simplest nonlinear operation is a multiplication of both affine forms:

$$f(\hat{a},\hat{b})=\hat{a}\cdot\hat{b}=a_0b_0+a_0b_1\varepsilon_b+a_1b_0\varepsilon_a+a_1b_1\varepsilon_a\varepsilon_b \tag{3}$$

Generalizing for any order, the centered form of the output polynomial is given by:

$$f(\hat{a},\hat{b})=\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}a_ib_j\varepsilon_a^i\varepsilon_b^j \tag{4}$$

It can be seen that this solution is an extension of AA in which all the high-order terms are taken into account [5]. In [21], this technique is just applied in the case of multiplications and other non-linear operations are obviated. Nevertheless, since the monomials of MAA are not orthonormal, the incorporation of the PCE techniques that take into account higher order terms when considering different types of operations is also required. Without them, the propagation of the gains throughout the system under analysis would not be accurately performed.

Polynomial Chaos Expansions (PCE)

Given a set of independent random variables of dimension *N*, $\Phi = \{\phi_1, \phi_1, \ldots, \phi_N\}$, and another random variable *Y*, square integrable, such that *Y* = *f*(Φ), then *Y* can be expressed as a weighted sum of polynomials as

$$Y = \sum_{i}^{\infty} \alpha_i \psi_i(\Phi) \tag{5}$$

where each α_i is a constant coefficient and each ψ_i is the *i*-th polynomial from an orthogonal basis [20]. The terms α_i are the spectral coefficients of the expansion, and the terms $\psi_i(\Phi)$ are the orthonormal polynomial basis, which satisfy the condition

$$<\psi_i, \psi_j> = \begin{cases} \psi_i^2 & if\ i = j \\ 0 & if\ i \neq j \end{cases} \tag{6}$$

In practice, the number of terms of the PCE is truncated to a finite number. It depends on the dimension of the expansion *n* (number of independent variables in vector Φ) and the maximum order of the polynomials used, *p*. The selection of the basis depends on the probability density functions (gaussian, uniform, gamma, beta, etc.) of the RVs present in the system. In particular, for the analysis of a given system with gaussian random variables, Hermite basis polynomials provide the most accurate results [22].

The coefficients of the expansion α_i in Eq. (5) are computed by applying a Galerkin projection operation [16, 19], and solved by applying Monte-Carlo techniques with a small number of samples.

Once the random input signals have been defined, and expressed as a function of the $\psi_i(\Phi)$ basis polynomials, the next step is to propagate the coefficients through the data flow graph. This procedure is exploits the orthogonality properties of the polynomials. The basic operations are performed as follows.

Consider two input RVs $\hat{x}$ and $\hat{y}$ expanded in a PCE,

$$\hat{x} = \sum_{i=1}^{m} x_i \cdot \psi_i\,,\ \hat{y} = \sum_{i=1}^{m} y_i \cdot \psi_i \tag{7}$$

The computation of the linear operations is straightforward, i.e.:

$$\hat{z} = a\hat{x} \pm b\hat{y} = \sum_{i=1}^{m} (ax_i \pm by_i) \cdot \psi_i \quad \Rightarrow \quad z_i = ax_i \pm by_i \tag{8}$$

The propagation through the non-linear operations such as the multiplication is not so direct. Considering that $\hat{z} = \hat{x} \cdot \hat{y}$ and substituting each variable by its correspondent PCE:

$$\hat{z} = \sum_{k=1}^{m} z_k \cdot \psi_k = \sum_{i=1}^{m} x_i \cdot \psi_i \sum_{j=1}^{m} y_j \cdot \psi_j \tag{9}$$

The coefficients z_k are calculated by performing a Galerkin projection [19]:

$$z_k = \sum_{i=1}^{m}\sum_{j=1}^{m}\frac{<\psi_i\psi_j\psi_k>}{\psi_k^2}x_i y_j = \sum_{i=1}^{m}\sum_{j=1}^{m}C(i,j,k)\; x_i y_j, \tag{10}$$

which constitutes a linear system of *m* equations. It can be expressed in matrix form as:

$$Z = A\cdot X, \text{ with } A = C\cdot Y \tag{11}$$

where *A* is an *m*×*m* matrix and X, Y and Z are the column vectors that correspond to the $\hat{x}$, $\hat{y}$ and $\hat{z}$ coefficients, respectively. Tensor *C*(*i*, *j*, *k*) is the same for a given dimension and order, so it only has to be calculated once (for instance in a pre-processing stage), and afterwards reused when needed, thus notably reducing the required computation time [16]. In addition, a number of techniques for accelerating the computation of the *C* matrix can be applied, speeding the overall process even further. The interested reader may find detailed examples of the propagation of affine forms using combined PCE + MAA in [16, 19].

Multi-Element generalized Polynomial Chaos (ME-gPC)

In many cases, PCE requires an excessively large basis to accurately represent the set of values. This happens particularly in the presence of discontinuities, or when many non-linear operations appear following each other. To overcome this, ME-gPC is formulated [WK05]. This technique partitions the input domain in smaller sub-domains, decomposing the complex functions into a set of simpler ones. This enables the efficient use of lower PCE orders to model the sub-domains, while still providing very accurate results [16].

Being *B* = [–1, 1]^n the domain in which Ξ = [ξ_1, ξ_2, …, ξ_n] is defined, the ME-gPC method proposes its decomposition in a regular set of non-overlapping elements. Each element will be now contained in the domain B_k = $[a_1^k, b_1^k)$ × $[a_2^k, b_2^k)$ × … × $[a_n^k, b_n^k]$, where a_i and b_i are respectively the upper and lower bounds of the *i*-th local random variable.

From this decomposition of the global domain, a local random vector for each element is now defined as ζ^k = [ζ_1^k, ζ_2^k, …, ζ_n^k]. Next, in order to take advantage of the properties of the Legendre Chaos, each ζ^k is re-scaled into a new random vector ξ^k = [ξ_1^k, ξ_2^k, …, ξ_n^k]. This vector is equivalent to ζ^k but in the domain [–1, 1]^n, instead of B_k.

Once a dimension has been partitioned, the new PCE expansions for each sub-domain are generated. Each of these expansions has the form

$$\tilde{u}(\tilde{\xi}) = \sum_{i=1}^{m}\tilde{u}_i\Phi_i(\tilde{\xi}) \tag{12}$$

where $\tilde{\xi}$ is defined [–1, 1]^d. To calculate the coefficients $\tilde{u}_i$ of each new expansion, a linear system of equations is solved. This system is generated by choosing *m*+1 uniform grid points $\tilde{\xi}_i$ in [–1, 1]^d.

With the expansions $\tilde{u}(\tilde{\xi})$ obtained with this method, PCE can be locally applied to the different elements. Once the expansions have been computed, the statistical global moments can be reconstructed applying Bayes' theorem and the law of total probability.

Figure 2 shows an example of the domain decomposition using ME-gPC for the conditional inequality $x^2 \geq y$.

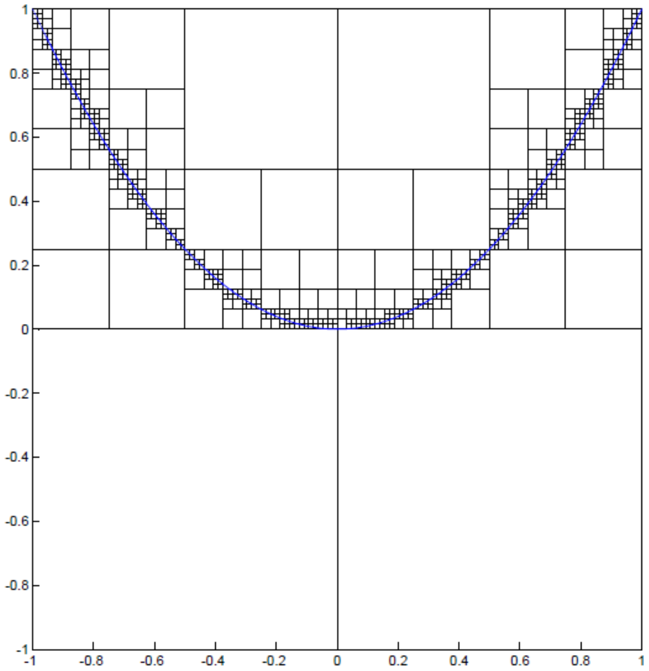


Figure 2. Example of domain decomposition using ME-gPC.

So far MEgPC has only been used to estimate the dynamic range in systems without control-flow structures, and it has been only applied to numerical procedures. In the following Sections we will combine MEgPC and MAA to estimate the sensitivity and the quantization noise in fixed-point digital systems with control-flow structures, extending the initial analysis carried out for linear systems in [6] to non-linear operations and control structures in the Data Flow Graph.

This largely broadens the applicability of the probabilistic interval analysis in word-length optimization, as it allows for an entire new class of systems to be targeted for modelling and optimization [16].

DSP Applications of the Extensions of Interval Computations

Some of the main applications of the Extensions of Interval Computations will be explained here, in different subsections, such as Dynamic Range Estimation, Quantization Noise, and Sensitivity Analysis in the different types of structures, including systems with discontinuities and control-flow structures.

The HOPLITE framework

In this Section a modular automated word-length optimization tool, HOPLITE, is introduced. One of its main objectives is to provide designers flexibility to perform modelling and search policies that best suit their objectives [16]. In the different subsections a general overview of the HOPLITE work flow will be provided, some of the implementation decisions, modules and interfaces of the framework, and a detailed execution example.

Table 2 provides a preliminary analysis of the languajes evaluated for its implementation, and Figure 3 shows a general overview of the functions included in the HOPLITE framework.

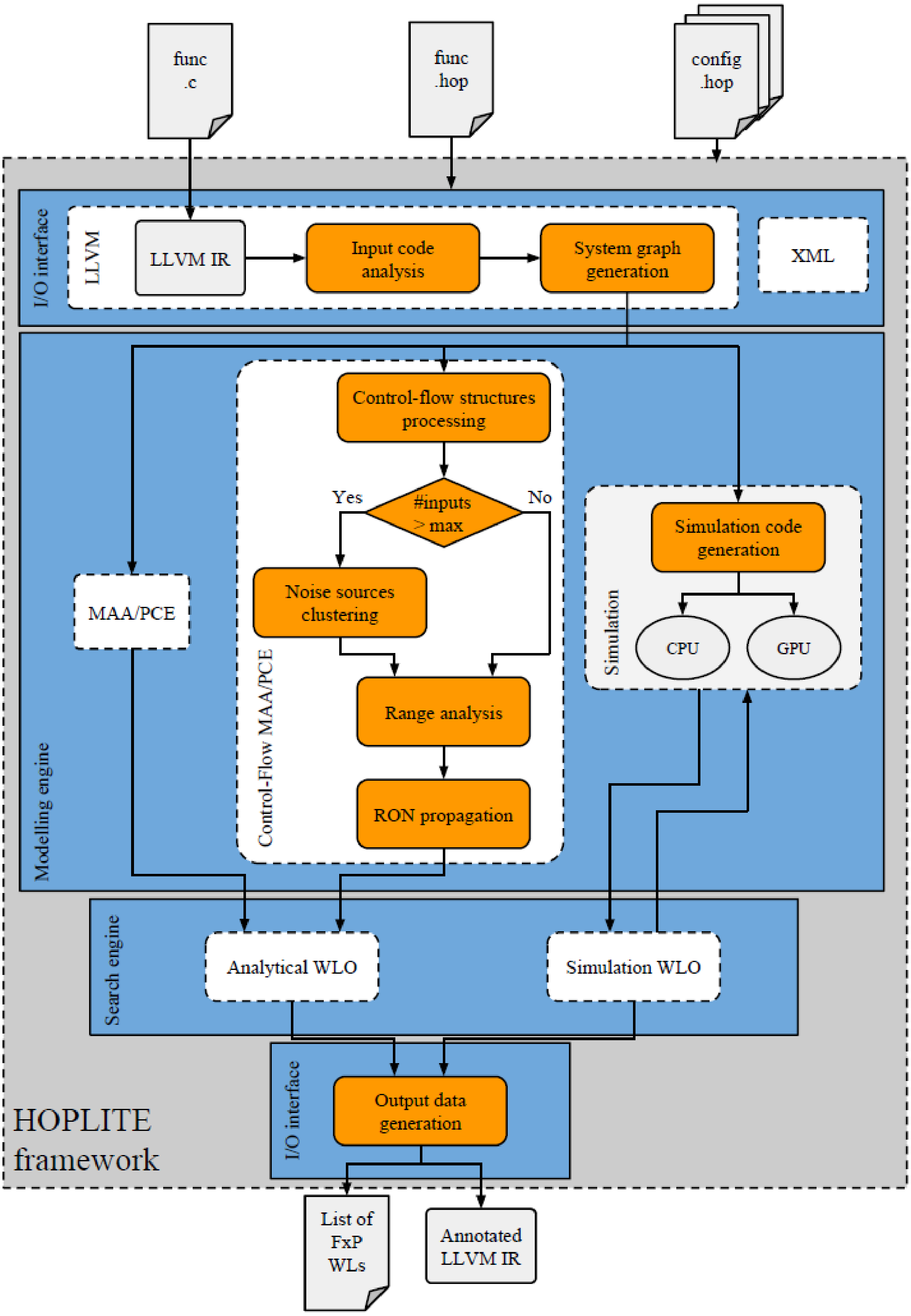
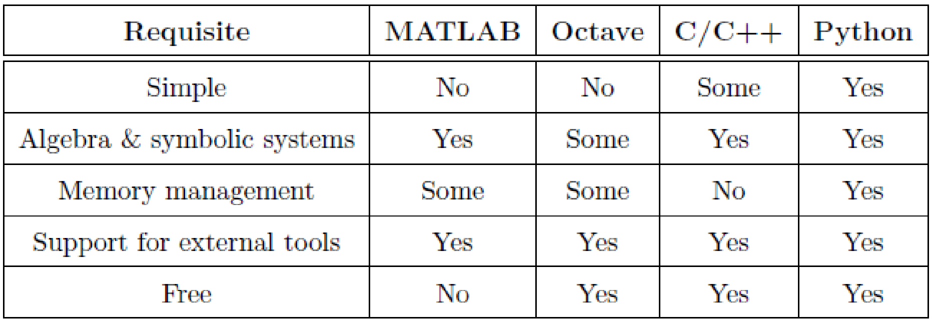


Figure 3. The HOPLITE framework work flow

Table 2. Language selection: requisites and availability



Requisite	MATLAB	Octave	C/C++	Python
Simple	No	No	Some	Yes
Algebra & symbolic systems	Yes	Some	Yes	Yes
Memory management	Some	Some	No	Yes
Support for external tools	Yes	Yes	Yes	Yes
Free	No	Yes	Yes	Yes

References

[1] M. Clark, M. Mulligan, D. Jackson, D. Linebarger. Accelerating fixed-point design for mb-ofdm uwb systems. *Comms Design,* EE times (online), 2005.

[2] R. Cmar, L. Rijnders, P. Schaumont, S. Vernalde, and I. Bolsens, “A Methodology and Design Environment for DSP ASIC Fixed Point Refinement,” in *Proc. conf. Design, automation and test in Europe, DATE ’99,* p. 56, 1999.

[3] K. I. Kum and W. Sung, “Combined Word-Length Optimization and High-Level Synthesis of Digital Signal Processing Systems,” *IEEE Trans. Circuits Syst.*, vol. 20, pp. 921–930, Aug. 2001.

[4] G. A. Constantinides, P. Y. K. Cheung, and W. Luk, “Wordlength Optimization for Linear Digital Signal Processing,” *IEEE Trans. Comput.- Aided Design Integr. Circuits Syst.*, vol. 22, n. 10, pp. 1432–1442, 2003.

[5] J. A. Lopez, C. Carreras, and O. Nieto-Taladriz, “Improved interval-based characterization of fixed-point lti systems with feedback loops,” *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 11, pp. 1923–1933, 2007.

[6] J. Lopez, G. Caffarena, C. Carreras, and O. Nieto-Taladriz, “Fast and accurate computation of the roundoff noise of linear time-invariant systems,” *IET Circuits, Devices and Systems,* vol. 2, no. 4, pp. 393–408, 2008.

[7] D. Menard, R. Rocher, and O. Sentieys, “Analytical Fixed-Point Accuracy Evaluation in Linear Time-Invariant Systems,” *IEEE Trans. Circuits and Systems I: Regular Papers,* vol. 55, November 2008.

[8] R. Rocher, D. Menard, N. Herve, and O. Sentieys, “Fixed-Point Configurable Hardware Components,” *EURASIP Journal on Embedded Systems,* vol. 2006, pp. Article ID 23197, 13 pages, 2006. doi:10.1155/ES/2006/23197.

[9] G. Caffarena, J. Lopez, G. Leyva, Carreras, and O. Nieto-Taladriz, “Architectural Synthesis of Fixed-Point DSP Datapaths Using FPGAs,” *Int. J. of Reconfigurable Computing*, vol. 2009, pp. 1–14, 2009.

[10] C. Fang, R. Rutenbar, and T. Chen, “Fast, accurate static analysis for fixed-point finite-precision effects in dsp designs,” in *Int. Conf. on Computer-Aided Design, 2003 (ICCAD ’03).* pp. 275–282, 2003.

[11] J. Lopez, Evaluacion de los Efectos de Cuantificacion en las Estructuras de Filtros Digitales Utilizando Tecnicas de Cuantificacion Basadas en Extensiones de Intervalos. PhD thesis, Univ. Politecnica de Madrid, Madrid, 2004.

[12] C. Shi and R. Brodersen, “Floating-point to fixed-point conversion with decision errors due to quantization,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP),* Montreal, 2004.

[13] S. Roy and P. Banerjee, “An algorithm for trading off quantization error with hardware resources for matlab-based fpga design,” *IEEE Trans.* Computers, vol. 54, no. 7, pp. 886–896, 2005.

[14] D.-U. Lee, A. Gaffar, R. Cheung, W. Mencer, O. Luk, and G. Constantinides, “Accuracy-Guaranteed Bit-Width Optimization,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 10, pp. 1990–2000, 2006

[15] L. Zhang, Y. Zhang, and W. Zhou, “Floating-point to fixed-point transformation using extreme value theory,” in *Eighth IEEE/ACIS Int. Conf. Computer and Information Science, 2009 (ICIS 2009).* pp. 271–276, 2009.

[16] E. Sedano, *Automated word-length optimization framework for multi-source statistical interval-based analysis of non-linear systems with control-flow structures*. PhD thesis, Univ. Politecnica de Madrid, Madrid, 2016.

[17] Y. Pang, K. Radecka, Z. Zilic. Optimization of imprecise circuits represented by taylor series and real-valued polynomials. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 29(8):1177-1190, 2010.

[18] D. Boland, G.A. Constantinides. A scalable precision analysis framework. *IEEE Trans. Multimedia*, 15(2), 242-256, 2013.

[19] L. Esteban. *High precision FPGA based phase meters for infrared interferometers fusion diagnostics.* PhD thesis, Universidad Politecnica de Madrid, 2011.

[20] B. Wu, J. Zhu, and F.N. Najm. Dynamic-range estimation. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 25(9), 1618-1636, 2006.

[21] H. Shou, H. Lin, R. Martin, G. Wang. Modified Affine Arithmetic Is More Accurate than Centered Interval Arithmetic or Affine Arithmetic. *Mathematics of Surfaces*, Lecture Notes in Computer Science, Springer. vol. 2768 pages 355-365. 2003.

[22] D. Xi,u G.E. Karniadakis. Modeling uncertainty in flow simulations via generalized polynomial chaos. *Journal of computational physics*, 187(1), 137-167, 2003.

[23] X. Wan, G.E. Karniadakis. An adaptive multi-element generalized polynomial chaos method for stochastic differential equations. *Journal of Computational Physics*, 209(2), 617-642, November 2005.