Expert Systems with Applications 42 (2015) 1189–1201

Contents lists available at ScienceDirect

Expert Systems with Applications

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

An evolutionary-based methodology for symbolic simplification of analog circuits using genetic algorithm and simulated annealing

Mohammad Shokouhifar *, Ali Jalali

Department of Electrical and Computer Engineering, Shahid Beheshti University, G.C., Tehran, Iran

ARTICLE INFO

Article history:
Available online 23 September 2014

Keywords:
Simplified symbolic analysis
Analog circuits
Modified nodal analysis
Multi-objective optimization
Genetic algorithm
Simulated annealing

ABSTRACT

In this paper, an evolutionary-based multi-objective criterion is introduced for simplified symbolic small-signal analysis of analog circuits containing MOSFETs. After circuit analysis via modified nodal analysis technique, the derived exact symbolic transfer function of the circuit behavior is automatically simplified. In contrast to traditional simplification criteria, the main objective of our criterion is to control the final simplification error rate. The proposed simplification methodology can be performed by such optimization algorithms as local-search algorithms, heuristic algorithms, swarm intelligence algorithms, etc. In this paper, a hybrid algorithm based on genetic algorithm and simulated annealing is applied to validate the proposed methodology. It is remarkable that all steps including netlist text processing, symbolic analysis, post-processing, simplification, and numerical analysis are consecutively derived in an m-file MATLAB program. The proposed methodology was successfully tested on three analog circuits, and the numerical results were compared with HSPICE.

1. Introduction

The aim of symbolic analyzers is to derive analytical characterization of the circuit behavior in terms of the circuit parameters, which are represented by symbols. In contrast to the numerical simulators like HSPICE, symbolic analyzers can generate symbolic expressions for the circuit behavior which are more instructive to designers. However, given a certain design point, a symbolic analyzer encounters higher computational complexity than a numerical simulator (Xu, Shi, & Li, 2011). Symbolic and numerical simulators should be viewed as complementary rather than competing tools. Numerical simulators serve to verify the performance of previously sized circuits, while symbolic tools serve to assist in predicting the behavior of unsized circuits (before sizing). The applications of modern symbolic tools can be basically grouped in two main areas: (1) Those associated with the generation of knowledge about the operation of circuits, e.g., insight into circuit behavior before sizing. (2) Those requiring repetitive evaluations of the formula describing the circuit characteristics, as in automated circuit sizing techniques via iterative optimization algorithms (Fernandez, Vazquez, Huertas, & Gielen, 1998).

Experience in symbolic analysis shows that the complexity of symbolic expressions grows exponentially with the circuit size, especially for the circuits described at device-level. For example, there is more than 4.5×10^{17} symbolic terms within the system denominator for the µA741 op-amp (Toumazou, Moschytz, & Gilbert, 2004). It is a serious problem in the practical use of these tools due to the difficulties of handling large symbolic formulas. However, experiments on practical circuits show that only a few terms in a symbolic expression contain the majority of relevant information of the circuit behavior (Fernandez et al., 1998). To deal with large analog integrated circuits, either simplification methods (Shokouhifar & Jalali, 2014) or hierarchical methods (Xu et al., 2011) must be applied. Hierarchical decomposition is to generate symbolic expressions in the "sequence-of-expression" forms. There are three methods for hierarchical analysis, namely topological analysis (Shi, 2013), network formulation (Hassoun & Lin, 1995), and DDD-based approaches (Tan, Guo, & Qi, 2005). The main drawback of all hierarchical-based exact symbolic analyses is that the generated sequence of expressions is difficult to interpret and manipulate (Tan, 2006). A number of research papers have addressed the simplified symbolic analysis techniques (Guerra, Roca, Fernandez, & Vazquez, 2002; Kolka, Biolek, Biolkova, & Dobes, 2011, 2012; Roo & Mazo, 2013; Shokouhifar & Jalali, 2014; Wambacq, Fernandez, Gielen, Sansen, & Vazquez, 1995; Yu & Sechen, 1996).

* Corresponding author. Tel.: +98 9121944288; fax: +98 2122431804.
E-mail addresses: m_shokouhifar@sbu.ac.ir, shokoohi24@gmail.com (M. Shokouhifar), a_jalali@sbu.ac.ir (A. Jalali).

http://dx.doi.org/10.1016/j.eswa.2014.09.030
0957-4174/

In general, the term *symbolic simplification* or *symbolic approximation* refers to hybrid symbolic-numeric algorithms used for the simplification of symbolic expressions, aim at minimizing the number of symbolic terms within the simplified expression while retaining high accuracy in representing the exact expression. These techniques require more numerical knowledge about the investigated circuit than manual simplifications do, but they yield compact expressions in a fully automated way. In manual simplification, the decision on which terms to keep and which ones to discard is only based on qualitative assumptions (e.g., $g_m r_o \gg 1$) that do not allow for assigning precise error to the simplified expressions. An automatic simplification algorithm is a computer program which needs an specific error criterion for simplifying a symbolic expression. On the other hand, qualitative assumptions are not sufficient for determining the relative importance of symbolic terms, especially when the expression consists of non-trivial combinations of symbols. Firstly, ISAAC (Gielen, Walscharts, & Sansen, 1989) and SYNAP (Seda, Degrauwe, & Fichtner, 1992), and later other tools like ASAP (Fernandez, Vazquez, & Huertas, 1991) and SSPICE (Wierzba et al., 1989) have introduced the idea of simplification.

Basically, simplification techniques can be categorized into three types: simplification-after-generation (SAG), simplification-during-generation (SDG) and simplification-before-generation (SBG) (Toumazou et al., 2004). In SAG methods, simplification is applied once the symbolic analysis has been performed and the exact expressions have been generated. Then, the simplified symbolic expression is constructed from pieces of the exact one. The main advantage of SAG techniques is that the simplification error rate can be controlled. However, symbolic analysis tools based on these methods, i.e. ISAAC, SYNAP, and SSPICE, were restricted to the circuits with maximum of 10 to 15 transistors (Toumazou et al., 2004). In SDG techniques, simplification is performed at the same time that the circuit is analyzed, that is, during the generation of symbolic expressions. Since these techniques do not generate the exact expressions, they are appropriate when the circuit size grows, and it is impossible to generate the exact expressions. The circuits with up to 25 transistors can be analyzed by these approaches (Toumazou et al., 2004). Such techniques have been implemented in SCYMBAL (Wambacq et al., 1995) and RAINIER (Yu & Sechen, 1996). In order to extend the capabilities of symbolic analysis tools even to the circuits with up to 40 transistors, the only possibility until today has been the utilization of the SBG techniques, which are performed on the circuit schematic, matrix, or graph before the symbolic analysis starts. Analog-Insydes (Sommer, Hennig, & Droge, 1993) and SIFTER (Hsu & Sechen, 1994) use SBG for simplification.

A circuit transfer function in expanded format has the general form as seen in Eq. (1), in which, the coefficients of s-powers are represented by sums-of-products of the symbolic parameters x. Generally, either $f_i(x)$ or $g_j(x)$ can be written as shown in Eq. (2), in which $h_{kt}(x)$ represents a product of symbolic parameters of the circuit, and is called a symbolic term. Polynomial h_k is the kth polynomial within the transfer function, which totally has l symbolic terms.

$$H(s,x) = \frac{\sum_{i=0}^{M}(s^i f_i(x))}{\sum_{j=0}^{N}(s^j g_j(x))} = \frac{f_0(x) + sf_1(x) + s^2 f_2(x) + \ldots + s^M f_M(x)}{g_0(x) + sg_1(x) + s^2 g_2(x) + \ldots + s^N g_N(x)} \quad (1)$$

$$h_k(x) = h_{k1}(x) + h_{k2}(x) + \cdots + h_{kl}(x) = \sum_{t=1}^{l} h_{kt}(x) \quad (2)$$

There are four common traditional criteria (Toumazou et al., 2004) for SAG. Partaking of the nominal values of the symbolic parameters, in all these criteria, simplification is performed separately on each polynomial within the numerator and denominator of the transfer function. The mathematical formulation of these criteria can be summarized in Table 1. In *Criterion1*, which has been used in SSPICE, the simplification of the polynomial h_k is as follows: At first, the term with the largest magnitude is found within the polynomial h_k, and is called h_{km}. Then, all terms within h_k are compared to h_{km}, one by one. According to Table 1, if the magnitude of term h_{kt} is smaller than $\varepsilon \times h_{km}$, it will be eliminated from the polynomial, in which ε is the user-specified maximum-allowed error tolerance for the simplification of each polynomial. The main drawback of this criterion is that the accumulated magnitude of the eliminated terms for each polynomial can be either a small or a large value in contrast to the total magnitude of the polynomial. If the polynomial h_k has a total of l terms, and in which the p terms among them satisfy *Criterion1*, the maximum generated error rate in contrast to the exact polynomial is $p \times \varepsilon$ for the worst case. As p grows exponentially with the circuit size, the generated simplification error could be larger than the user-specific value. In order to overcome the mentioned drawback, three other criteria were introduced. In *Criterion2*, in general, p terms can be eliminated from the polynomial h_k, if the absolute value of the accumulated magnitudes of the eliminated terms does not deviate from a given threshold. The denominator of *Criterion3* is identical with the previous one, however, the sum of the magnitudes of the eliminated terms is u in the numerator. *Criterion4* shares the numerator of the *Criterion3*, differing from it only in terms of the fact that the accumulate value of the magnitudes of all terms is calculated for its denominator.

As mentioned above, simplification in these traditional criteria was performed separately on each polynomial within the exact symbolic transfer function. Therefore, these criteria do not guarantee the accuracy of the final simplified symbolic transfer function. On the other hand, although the maximum error tolerance for the simplification of each polynomial is limited by ε, the final generated simplification error could not be controlled (e.g., in terms of magnitude, phase, poles, zeros, etc). Although these traditional criteria are well-known and easy to implement, they might lead to generating high error rates in simplified expressions. In order to overcome this disadvantage, we propose a new multi-objective SAG criterion for simplification, which considers some concepts from the overall transfer function to simplify it. In this method, the correlation between the polynomials of the transfer function is also considered to simplify them. The proposed criterion in this study guarantees the accuracy of the simplified symbolic expressions in contrast to the exact ones, with a predictable error rate. Recently, we have proposed an ant colony optimization algorithm for the simplification of symbolic transfer functions of analog circuits, which considers the mean-square error in gain/phase and the absolute error in gain/phase margins between the exact symbolic expressions and the simplified ones, for evaluation of artificial ants (Shokouhifar & Jalali, 2014). The proposed multi-objective criterion in this paper considers more concepts than in Shokouhifar and Jalali (2014) to simplify the symbolic expressions (e.g., the position of poles/zeros, dc-gain, unity gain-bandwidth frequency, etc).

The simplification problem is a binary selection problem to find an optimal subset from the set of all original symbolic terms. The binary subset selection techniques can be categorized in

Table 1
Comparison of the four traditional simplification criteria.

Criterion1	*Criterion2*	*Criterion3*	*Criterion4*
$\frac{\lvert h_{kt}\rvert}{\lvert h_{km}\rvert} < \varepsilon$	$\frac{\left\lvert\sum_{m=1}^{p} h_{km}\right\rvert}{\left\lvert\sum_{t=1}^{l} h_{kt}\right\rvert} < \varepsilon$	$\frac{\sum_{m=1}^{p} \lvert h_{km}\rvert}{\left\lvert\sum_{t=1}^{l} h_{kt}\right\rvert} < \varepsilon$	$\frac{\sum_{m=1}^{p} \lvert h_{km}\rvert}{\sum_{t=1}^{l} \lvert h_{kt}\rvert} < \varepsilon$