A Regularity-based Hierarchical Symbolic Analysis Method for Large-scale Analog Networks *

Alex Doboli and Ranga Vemuri
Digital Design Environments Laboratory
ECECS Department, University of Cincinnati
Cincinnati, OH, 45221-0030
{adoboli,ranga}@ececs.uc.edu

1. Extended Abstract

The main challenge for any symbolic analysis method is the exponential size of the produced symbolic expressions [2] (10^{11} terms for an op amp [1]). Current research considers two ways of handling this limitation: approximation of symbolic expressions and hierarchical methods. Approximation methods [2] retain only the significant terms of the symbolic expressions and eliminate the insignificant ones. The difficulty, however, lies in identifying what terms to eliminate and what the resulting approximation error could be. Hierarchical methods [1] tackle the symbolic analysis problem in a *divide-and-conquer* manner. They consider only one part of the global network at a time and then recombine partial expressions for finding overall symbolic formulas. Existing hierarchical methods have a main limitation in that they are not feasible for addressing networks that are built of tightly coupled blocks i.e. operational amplifiers [2].

The originality of our research stems from exploiting *regularity aspects* for addressing the exponential size of produced symbolic expressions. As a result, polynomial-size models are obtained for any network, including networks formed of tightly coupled blocks. Two kinds of regularity aspects were identified:

- *Analog networks have structural regularity*: A filter can consist of multiple stages of blocks (adders, integrators) that are linked in identical templates, that we call *Generic Interconnection Templates* (GIT). GITs can also be identified among the transistors of an operational amplifier. If a *generic* symbolic expression describes a GIT then this expression can be re-used for all blocks connected in this template. This significantly reduces the size of our symbolic models. Moreover, exploiting structural regularity is not limited to networks built of loosely coupled stages.
- *Symbolic expression regularity*: Symbolic parameters of a GIT are expressed by applying the same set of operations to distinct variables. The uniform set of operations is encapsulated in a *generic* symbolic function and re-used for expressing all parameters of a GIT.

This research discusses a novel hierarchical method for top-down symbolic analysis of large analog networks. The proposed technique includes three specific components:

- An *efficient representation of symbolic expressions*: The representation *Analog Performance Tree* (APT) explicitly targets the expression of hierarchy and regularity aspects of symbolic expressions. It is an uninterpreted variant of the closed-form, symbolic expressions that are created by traditional methods.
- An *algorithm for constructing APTs for symbolic expressions*: The method traverses top-down the hierarchy of a network and builds symbolic expressions depending on the GITs that were found, without explicitly solving nodal equations. The algorithm uses the hierarchy and regularity structure of a network for producing compact symbolic expressions.
- A *technique for extracting the structural regularity of a network*: We propose a decomposition method that identifies frequently used GITs. This is different from other partitioning approaches [1] for hierarchical symbolic analysis in that ours can handle tightly connected networks.

For large networks, the size of the symbolic models produced by our method is much less than the size of the models produced by other methods such as the two-graph method.

*This work was sponsored by the USAF, Air Force Research Laboratories, Wright Patterson Air Force Base under contract number F33615-96-C-1911

References

[1] M. Hassoun, P.-M. Lin, "Hierarchical Network Approach to Symbolic Analysis", *IEEE Trans. on Circuits and Systems*, April 1995.

[2] P. Wambacq *et al*, "Symbolic Network Analysis Methods for Practical Analog Integrated Circuits: A Survey", *IEEE Trans. on Circuits and Systems II*, October 1998.