

Topological Approach to Symbolic Pole-Zero Extraction Incorporating Design Knowledge

Guoyong Shi, *Senior Member, IEEE*

Abstract—This paper addresses the problem of automatic analytical pole-zero extraction for multi-stage operational amplifiers with frequency compensation. Traditional methods mainly rely on numerical reference to derive approximate pole-zero expressions without incorporating any design knowledge. Such methods suffer from bad interpretability of the auto-generated results. This paper takes a topological approach and attempts to advocate that certain form of design knowledge can be incorporated in the symbolic term selection process for pole-zero generation. The generation engine selects the dominant terms by a formal inspection on the token patterns that are correlated to gain factors and compensation elements. Since the gain factors and compensation elements of an opamp are pertinent to the topological details of a circuit, the proposed pole-zero extraction method is closer to design conception than other numerical reference based methods. Consequently, the generated pole/zero results are better interpretable. Application to a class of multi-stage operational amplifiers with a variety of compensation structures demonstrates that the proposed method is effective and can match human-derived results.

Index Terms—analog design automation, graph-pair decision diagram (GPDD), pole and zero (PZ), operational amplifier (opamp), symbolic analysis.

I. INTRODUCTION

In the design of analog integrated circuit (IC) cells frequency compensations in the forms of feedback and/or feed-forward paths are commonly used. A variety of compensation strategies have been studied in many multi-stage operational amplifier (opamp) design works [1], [2], [3], [4], [5]. To ensure that the closed-loop circuit has sufficient stability margins, the pole-zero (PZ) properties of the transfer function must be examined before proceeding to sizing and biasing. For the traditional one- or two-stage opamps, the pole-zero analysis is relatively simpler and can be carried out by manual analysis. When multi-stage designs are considered, compensation strategy becomes more intricate and analysis gets more involved.

Technology advancement drives device feature size to be much smaller and supply voltages to be further lower. Designers are thus forced to consider multi-stage designs to realize opamps with much higher gain and better signal swing. Recently, research contributions to multi-stage opamp design are quickly rising [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16]. For opamps involving more than two stages, manual transfer function analysis becomes less wieldy. Designers have to manually derive all terms by solving nodal equations

or using signal-flow graph based analysis. Then designers use their design knowledge to screen the long list of terms and prune those less significant terms. This process of term selection highly relies on the human knowledge on the analog design practice, which is nontrivial for the beginners.

However, deriving closed-form pole-zero (PZ) expressions is still considered a necessary analytical step in the contemporary design practice. This is because several crucial design steps like compensation, stability, and other frequency-performance related reasoning must be based on the closed-form PZ expressions. A key step toward PZ derivation is to generate a simplified transfer function that captures the circuit frequency response. Therefore, the main work for symbolic PZ generation is to construct a simplified symbolic transfer functions in which only the dominant coefficient terms are retained. A basic requirement on the usefulness of automatically generated PZ expressions is that such closed-form results should be readable or interpretable, so that based on such results designer can proceed to reason on pole-zero relocation, cancelling, and resolving conflicts, etc. Many frequency compensation related design techniques are discussed in the survey paper [2].

In order to facilitate design space exploration and speed up the analytical design process, a tool capable of automatic pole-zero generation would be of great help. To this date, numerous methods have been proposed for the problem of automatic pole-zero extraction. However, most of them cannot successfully serve the design need due to missing of certain critical ingredients that are commonly involved in human reasoning.

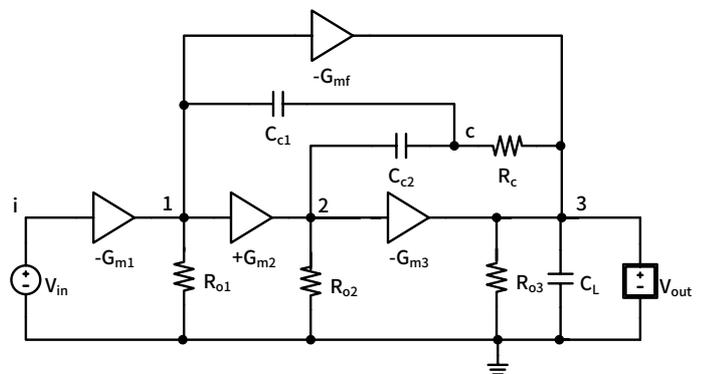


Fig. 1. Three-stage opamp with nested Miller compensation and one nulling resistor (NMCNR) [10].

Show in Fig. 1 is a three-stage opamp with frequency compensation containing two Miller capacitors C_{c1} and C_{c2} , one nulling resistor R_c , and one feedforward gain element G_{mf} . Such a compensation structure is typically called the

This research was supported by the National Natural Science Foundation of China (NSFC) under grants No. 61176129 and No.61474145.

G. Shi is with the Department of Micro/Nano-electronics, Shanghai Jiao Tong University, Shanghai, China (e-mail: shiguoyong@sjtu.edu.cn).

©Copyright (c) 2017 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending an email to pubs-permissions@ieee.org

nested Miller compensation (NMC) [10]. Stability analysis on this conceptual circuit structure is crucial to the later stages of design where a transistor-level CMOS circuit is to be constructed and sized.

Many manual methods can be used to derive the transfer function from V_{in} to V_{out} for the given circuit in Fig. 1. To have a feeling on how many terms one has to deal with, we list all the terms of the voltage transfer function explicitly below which were generated by a symbolic analysis tool. We shall arrange the transfer function in the following form

$$T(s) = \frac{N(s)}{D(s)} = \frac{b_0 + b_1s + \dots + b_ms^m}{a_0 + a_1s + \dots + a_ns^n}. \quad (1)$$

For the current example the numerator polynomial $N(s)$ is of the second degree and the denominator polynomial $D(s)$ the third degree, whose coefficients are listed below:

$$\begin{aligned} b_0 &= R_{o2}R_{o3}R_{o1}\mathbf{G}_{m1}\mathbf{G}_{m2}\mathbf{G}_{m3} + R_{o3}R_{o1}\mathbf{G}_{m1}\mathbf{G}_{mf} \\ b_1 &= R_{o2}R_{o3}R_{o1}\mathbf{G}_{m1}C_{c2}\mathbf{G}_{mf} + \mathbf{R}_cR_{o3}R_{o1}\mathbf{G}_{m1}C_{c2}\mathbf{G}_{mf} \\ &\quad + \mathbf{R}_cR_{o2}R_{o3}R_{o1}\mathbf{G}_{m1}\mathbf{G}_{m2}\mathbf{G}_{m3}C_{c2} \\ &\quad + \mathbf{R}_cR_{o2}R_{o3}R_{o1}C_{c1}\mathbf{G}_{m1}\mathbf{G}_{m2}\mathbf{G}_{m3} \\ &\quad + \mathbf{R}_cR_{o3}R_{o1}C_{c1}\mathbf{G}_{m1}\mathbf{G}_{mf} - R_{o2}R_{o3}R_{o1}\mathbf{G}_{m1}\mathbf{G}_{m2}C_{c2} \\ &\quad - R_{o3}R_{o1}C_{c1}\mathbf{G}_{m1} \\ b_2 &= \mathbf{R}_cR_{o2}R_{o3}R_{o1}C_{c1}\mathbf{G}_{m1}C_{c2}\mathbf{G}_{mf} \\ &\quad + \mathbf{R}_cR_{o2}R_{o3}R_{o1}C_{c1}\mathbf{G}_{m1}\mathbf{G}_{m3}C_{c2} \\ &\quad - R_{o2}R_{o3}R_{o1}C_{c1}\mathbf{G}_{m1}C_{c2} \end{aligned} \quad (2)$$

and

$$\begin{aligned} a_0 &= 1 \\ a_1 &= R_{o1}C_{c1} + R_{o2}R_{o3}R_{o1}C_{c1}\mathbf{G}_{m2}\mathbf{G}_{m3} + R_{o3}R_{o1}C_{c1}\mathbf{G}_{mf} \\ &\quad + R_{o2}C_{c2} + \mathbf{R}_cC_{c2} + R_{o3}C_L + R_{o3}C_{c2} \\ &\quad + R_{o2}R_{o3}\mathbf{G}_{m3}C_{c2} + \mathbf{R}_cC_{c1} + R_{o3}C_{c1} \\ a_2 &= -\mathbf{R}_cR_{o2}R_{o1}C_{c1}\mathbf{G}_{m2}C_{c2} - R_{o2}R_{o3}R_{o1}C_{c1}\mathbf{G}_{m2}C_{c2} \\ &\quad + R_{o2}R_{o1}C_{c1}C_{c2} + \mathbf{R}_cR_{o1}C_{c1}C_{c2} + R_{o3}R_{o1}C_{c1}C_L \\ &\quad + R_{o2}R_{o3}R_{o1}C_{c1}C_{c2}\mathbf{G}_{mf} + R_{o3}R_{o1}C_{c1}C_{c2} \\ &\quad + R_{o2}R_{o3}R_{o1}C_{c1}\mathbf{G}_{m3}C_{c2} + R_{o2}R_{o3}C_LC_{c2} + \\ &\quad \mathbf{R}_cR_{o3}C_LC_{c2} + \mathbf{R}_cR_{o2}C_{c1}C_{c2} + \mathbf{R}_cR_{o3}C_{c1}C_L \\ &\quad + R_{o2}R_{o3}C_{c1}C_{c2} \\ a_3 &= -\mathbf{R}_cR_{o2}R_{o3}R_{o1}C_{c1}C_L\mathbf{G}_{m2}C_{c2} \\ &\quad + R_{o2}R_{o3}R_{o1}C_{c1}C_LC_{c2} \\ &\quad + \mathbf{R}_cR_{o3}R_{o1}C_{c1}C_LC_{c2} + \mathbf{R}_cR_{o2}R_{o3}C_{c1}C_LC_{c2} \end{aligned} \quad (3)$$

where the constant term a_0 of the denominator has been normalized.

In principle the closed-form pole-zero expressions can be derived from the s -coefficients listed above. However, designers would not do this way. Typically they would pick those so called *dominant* terms that are of concern in design reasoning. Design knowledge is essential when a designer selects the terms to be included in the PZ expressions. During this process one thing is certain: designer would not pick terms by attempting to evaluate them based on a set of reference element values.

Then the following questions arise: 1) What kind of design knowledge that a designer would use in picking those significant terms? 2) Whether such design knowledge can be incorporated in a CAD tool?

Roughly speaking, the following term selection guides make sense:

- Since the main goal of an opamp is to achieve gain, terms containing more factors of G_mR symbolize higher significance.
- Nulling resistors like R_c require special attention because they are the elements that designers would use to relocate zeros or enable pole-zero cancellation.

However, rules as stated above can be carried out easily by human, but not by a computer. They lack quantitative nature in order to be incorporated in an automation tool. We need to examine whether the descriptive statements can be formulated into token-based quantitative measures so that terms can be differentiated by assessing such deterministic measures. This is the main research task of this paper, and to this end we shall introduce a symbolic term generation paradigm that can perfectly incorporate the token-based term selection scheme.

This paper is organized as follows. We first make an overview in the next section on the related works that have dealt with automatic pole-zero extraction and point out their limitations or incapability to handle the needs we have outlined above. In section III we make an outline on the topological pole-zero extraction framework, where design intuition is emphasized. Then in section IV we propose term selection criteria (mainly token count inequalities) that can be implemented as post-processing tasks inside the graph-pair decision diagram (GPDD) algorithm [17]. Applications of the proposed method to a set of eleven multi-stage opamps are reported in section V. The auto-generated results are compared to those human-derived closed-form results reported in several solid-state circuit publications to validate the effectiveness. This paper is concluded in section VI.

II. RELATED WORKS

Automatic pole-zero extraction in analytical form for an analog circuit cannot be done by running a numerical SPICE simulator. Symbolic circuit analysis is the only viable approach to this problem. Since the early days of research on symbolic circuit analysis tools, pole-zero extraction has been always a targeted application of a symbolic tool [18], [19], [20]. However, unlike generating exact symbolic transfer functions, whose accuracy or correctness can be verified by a SPICE simulation tool or other existing symbolic analysis tools, pole-zero generation is very different. The main reason is due to the fact that no standard analytical expressions exist for the poles and zeros of most circuits, except for some small circuit examples commonly studied in textbooks [21], [22]. In practice poles and zeros are commonly derived by designers manually. While deriving, designers often drop insignificant terms based on their design conception. Application need is often an important factor in guiding designers to make the term selection decision. Most of the time the key guide is the relative significance between several terms in summation. As

long as certain terms are relatively insignificant comparing to others, designers may rule out such terms for consideration. This step is mainly for collecting design insight, rather than for validating circuit performance.

So far, most of the existing works on automatic pole-zero generation in closed-form use one kind or another numerical reference based method, in which incorporating human design knowledge in the term selection process is uneasy. Some works even adopted sophisticated numerical computational algebra for term selection, completely ignoring the fact that those dominant terms ought to be retained carry circuit meaning that can be inferred in the context of design knowledge. We discuss in more details below on the limitation of those computational strategies proposed in the literature.

The work by Wambacq et al. [23] proposed to apply a symbolic simulator in practical circuit design. The main goal of this work was to extend the capability of a symbolic simulator to being able to analyze larger size circuits (targeting at analog cells consisting of 20 transistors or more). In this short note a technique called *simplification during generation* (SDG) was introduced, but the technique was not specifically aimed at pole-zero extraction, although the ideology was applied later by other authors to the practice of pole-zero extraction. It is worth noting that the SDG technique requires numerical reference and an error control strategy. The SDG strategy is sometimes problematic when terms are correlated and the numerical reference values for the compensation elements are not decided a priori.

Another representative work published in the solid-state circuit design community was the paper by Nebel et al. [24]. This work presented the SANTAFE tool for symbolic pole/zero calculation. This tool integrates two key ingredients: signal-flow graph (SFG) for symbolic transfer function generation and *symbolic Newton-iteration* for correcting the poles and zeros generated by the root splitting technique. Because each Newton-iteration step involves multiple algebraic manipulation steps (addition, multiplication, and division, following a differentiation step), the complexity of the intermediate expressions grows quickly. Therefore, intermediate term-pruning is mandatory in order to control the complexity of expressions. Terms being pruned are mainly based on their relative numerical values given numerical reference. Correlations among the terms and design knowledge within the terms are completely ignored. Another issue is that frequency compensation was not explicitly considered in the applications attempted in [24].

The work by Guerra et al. [25] is based on the formulation of the time-constant matrix of a circuit. The circuit matrix is formulated by the Modified Nodal Analysis (MNA) method. The *modification-decomposition* technique proposed by Haley [26] was applied for pole-zero extraction. *Modification* refers to a procedure of down-dimensioning the admittance matrix by transforming the nodal voltage state space into the subspace defined by the capacitors. The resulting reduced-dimensional matrix is the *time-constant matrix*. *Decomposition* refers to the subsequent QR factorization step. QR factorization is a well-known numerical method for computing matrix eigenvalues. The work by Guerra et al. mainly used it for generating numerical pole/zero reference in purpose of guiding the term

simplification steps in the phase of symbolic manipulation.

There are more works that follow the similar line of numerical mathematics treatment, see [27], [28], [29] among others. Because this line of research mainly deals with the pole-zero extraction problem as a reformulated numerical computation problem rather than a circuit-based inference problem, the generated symbolic results run into problems like lack of intuitiveness and interpretability. On the other hand, certain practical needs such as considering the compensation elements explicitly in pole-zero generation cannot be treated easily. The reason is that designers do not have an a priori specification on the numerical values of the compensation elements at an initial design stage.

The simplification techniques proposed by Hsu and Sechen [30] are based on dividing the frequency axis into several sub-ranges, within each range a small portion of poles or zeros are isolated. Simplification is performed at the matrix level by sorting the matrix entries according to their numerical values and their effect on the magnitude and phase of a transfer function. The frequency response of a circuit is highly dependent on the circuit details. Before a circuit is well-designed and sized, partitioning a frequency range into several sub-divisions for pole-zero extraction would not be a reliable extraction strategy. Moreover, as observed by [25], this method may miss certain cancelling pole-zero pairs.

The method proposed by Daems et al. in [31] is based on graph decomposition techniques. The overall signal-flow graph representation of a circuit is decomposed into smaller portions according to the location of poles and zeros. This method also requires partitioning the frequency range into several subsets in order to localize poles and zeros to facilitate the efficiency of extraction. Hence, the extraction efficiency depends on a priori knowledge on the pole/zero distribution. Another limitation of the method is that the implementation of SFG-based symbolic analysis requires involved data structures in term generation and the generated terms are not guaranteed to be cancellation-free. Design knowledge perceivable from terms might be dimmed by the complicated graph-level manipulations.

We now make a summary on the limitations of the existing methods:

- The symbolic simplification steps employed by most of the existing works lack a circuit-based algorithmic design that can incorporate design knowledge.
- All works require a certain kind of numerical reference for pole-zero generation. Numerical reference based extraction blurs the circuit meaning of the generated poles/zeros.
- Some of them introduce highly involved algebraic manipulation steps in pole-zero generation. Because such steps are not reversible, it prevents analog designers from appreciating the circuit meaning of the generated results.
- Most of the introduced error control schemes are mainly out of tool developer's decision-making, such decisions might not fully consider inter-term correlation and the validity of design reasoning.
- All numerical reference methods require repeated numerical evaluations for pruning each term. This could be highly costly when the pole-zero generation process

starts from a transfer function containing millions of terms (which is possible if starting from a transistor-level circuit).

- Term-based pruning is the main cause of losing correlations existing among the terms. Building explicit term correlation is important in making pole-zero compensations in design.
- None of them provide strategies for considering the effect of compensation on pole-zero extraction.
- None of them have considered incorporating design knowledge in a pole-zero extraction process.

III. PROPOSAL OF A TOPOLOGICAL FRAMEWORK ENABLING INCLUSION OF DESIGN KNOWLEDGE

We propose a topological framework for pole-zero extraction, where by *topological* we mean that the macro-level circuit topology is maintained throughout the process of pole-zero extraction. Fig. 2 shows the key stages included in the framework. If poles and zeros need to be extracted for an existing transistor-level circuit, then the flow would start from *Stage 1*. The transition from Stage 1 to Stage 2 is essentially a topological circuit simplification step, within which the macro-level circuit topology shall be preserved.

There are several reasons that justify the adoption of Stage 2 in the framework. Firstly, it follows the design convention. Most design practice starts circuit design from a macro-level (or conceptual level) circuit, which serves as the backbone architecture that the final transistor-level implementation should be confined to. Secondly, directly deriving poles and zeros for a transistor-level circuit is less intuitive and convenient. Poles and zeros derived from a macro-level circuit can offer better design insight in terms of developing compensation strategy and posing sizing constraints. Thirdly, there already exists a viable methodology to automatically generate a macro-level circuit from a transistor-level one by preserving the macro structure (see [32] and its extended version [33]).

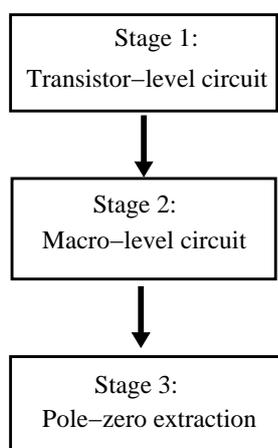


Fig. 2. The pole-zero extraction flow adopted in this work.

Hence, in a sense we have provided two entrances to the pole-zero extraction framework; either by entering the tool from Stage 1 given a transistor-level circuit, or by entering

from Stage 2 given a macro-level circuit. This conceptual tool design is for the purpose of tool flexibility and adheres to the design practice.

The main difference between our work and those existing ones is that we treat the poles and zeros of a circuit as an inherent property pertinent to the circuit topology. Throughout the extraction process, we do not formulate any intermediate matrix representations. Hence, no algebraic simplification would be needed. This is the key contributing factor that can render the generated poles and zeros better readable because the extraction process is based on rules rather than on numerical computation.

We further stress that by our approach both transitions between the three stages shown in Fig. 2 are based on manipulating the circuit topology. The underlying computation technology is the GPDD theory developed in [34], which will be briefly reviewed in the next section. The GPDD algorithm creates a shared BDD (binary decision diagram) data structure [35]. Hence, unlike the earlier methods, terms are not treated as individual standalone objects, rather they are interleaved by the symbolic term saving data structure (i.e., a BDD with path sharing). Partial sub-terms can be shared automatically in common sub-paths in BDD without extra cost by human manipulation. When a variable (or a circuit element) is considered for pruning, a subset of the correlated terms are processed altogether. The topological circuit reduction procedure developed in [32] was also based on the fundamental BDD-based term-sharing property.

In other words, the circuit simplification phase in our method is not by pruning terms, rather it is by pruning the circuit structure. Although numerical reference is also needed by the topological reduction procedure proposed in [32], [33], error control becomes much simpler and more robust because the method is based on monitoring only a few critical frequency response points during circuit reduction. We once again stress that during this stage we do not extract poles and zeros.

After the circuit reduction, a macro-level circuit like the one shown in Fig. 1 can be generated. Then the pole-zero extraction procedure begins to work. Since the reduced macro-level circuit is symbolic in that all of the model parameters are expressed in terms of the small-signal parameters in the original transistor-level circuit, we do not lose the connection between the extracted pole-zero results (in the third stage) and the original transistor-level circuit (in the first stage). This benefit is not offered by the other methods we reviewed earlier.

Since the proposed symbolic deduction flow is fully topological, incorporating design knowledge in the work flow becomes much easier. First of all, working on a macro-level backbone circuit structure (i.e., a stage-form circuit model with intact compensation structure) is itself a utilization of design knowledge. Such a circuit manipulation strategy is completely different from other proposals that require MNA matrix manipulation or signal-flow graph based deduction. Secondly, starting from a macro-level circuit it is possible to further incorporate other detailed design knowledge in the process of pole-zero extraction. For example, we may impose conditions on retaining those terms that contain the highest

number of gain factors.

IV. GPDD AS THE FOUNDATION FOR TERM SELECTION

A. GPDD Basics

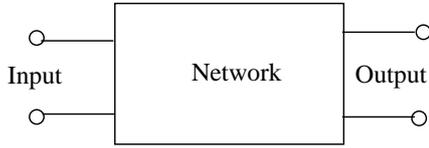


Fig. 3. Linear 2-port network with one input and one output.

Given a 2-port linear network with specific input and output as shown in Fig. 3, we may use a symbolic analysis method to derive its analytical transfer function. GPDD is a binary decision diagram that can be used to derive a symbolic transfer function by reducing the network (represented as a pair of graphs) until the circuit becomes a trivial single node. While in reduction, sharable graphs are identified and shared in the BDD. After the construction finishes, the paths of BDD (starting from the root and terminating at the vertex *One*) represent the symbolic terms. Due to the sharing property of BDD, the algebraic terms are stored with maximal sharing, rather than being enumerated one after another [17].

Before GPDD construction, we define the input-output relation as a dependent source, with the output controlling the input, i.e., $u_{in} = Xu_{out}$, where the variable u stands for a current or a voltage. Apparently, X is the gain factor of a dependent source of type E, F, G, or H. (In terms of the SPICE syntax, E, F, G, and H refer to VCVS, CCCS, VCCS, and CCVS elements, respectively.) GPDD can analyze circuits consisting of R, L, C, E, F, G, H, and nullor elements, encompassing all small-signal elements commonly encountered in analog integrated circuit modeling.

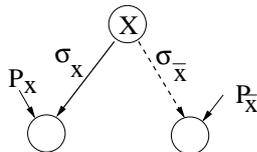


Fig. 4. The triple vertices at the GPDD root.

As a BDD, GPDD is composed of a set of vertex triples connected in the form shown in Fig. 4. Unlike a BDD representing logic functions [35], GPDD has signed directed links (1-arrows and 0-arrows), where the signs along a path are combined to determine the term *sign*. After all such vertex triples are connected by a top-down recursion, a depth-first GPDD traversal can scan all the symbolic terms. As a convention, we always place the I/O symbol X at the root by construction so that the whole set of the terms generated by the GPDD can be split into two complementary subsets, one containing all terms with X while the other not.

By running bottom-up recursion once, we obtain the following equation at the GPDD root

$$P(s) := X \cdot \sigma_x \cdot P_x(s) + \sigma_{\bar{x}} \cdot P_{\bar{x}}(s) = 0, \quad (4)$$

where $P_x(s)$ and $P_{\bar{x}}(s)$ denote respectively the s -polynomials retrieved at the vertexes pointed by P_x and $P_{\bar{x}}$, and σ_x and $\sigma_{\bar{x}}$ denote respectively the signs attached to the two pointers. Solving X from (4) gives rise to the symbolic transfer function $H(s)$ expressed by the reciprocal of X ; namely,

$$H(s) = \frac{1}{X} = -\frac{\sigma_x \cdot P_x}{\sigma_{\bar{x}} \cdot P_{\bar{x}}}. \quad (5)$$

B. Term Properties for Multi-stage Amplifiers

Now we proceed to consider a special type of circuits that consist of multiple transconductance stages in the abstract form depicted in Fig. 5. In principle, the number of stages can be extended in cascade.

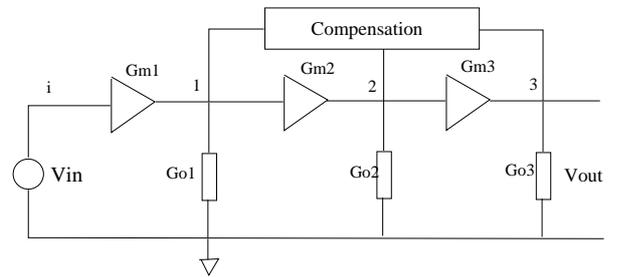


Fig. 5. Abstract schematic of a multi-stage opamp.

A graph description of the above circuit is plotted in Fig. 6, where to be specific the compensation structure of Fig. 1 is assumed (but for simplicity we temporarily omit the feedforward element G_{mf}). Elements G_{o1} , G_{o2} , and G_{o3} represent the output admittances of all stages. In general, each output admittance takes the form of $G_{oi} = R_{oi}^{-1} + sC_{oi}$, where R_{oi} and C_{oi} are the output resistance and capacitance of the i th stage, where $i = 1, 2, 3$.

To continue, we need to borrow some terminologies from [17] to facilitate the description. The graph edges are marked with their names. Depending on the nature of each edge (R, G_m , VCCS, or VCVS), appropriate names are coined and attached to the edges. For example, the edges named by C_{c1} , C_{c2} , R_c , G_{o1} , G_{o2} , and G_{o3} in Fig. 6 are self-evident. The edges named by VC1 and CS1 represent the transconductance element of G_{m1} , which is a VCCS. The other edges of VC2, CS2, VC3, and CS3 are similar. The remaining VC and VS edges represent the I/O relation of the opamp. The input and output edges (named by VC and VS) are included in the graph as a pair of dependent source as well. The edges are connected at their corresponding ports defined by the original circuit schematic in Fig. 5. It is clear that all edges seen in the graph have their corresponding circuit branches as indicated by the edge names.

The GPDD algorithm is based on the two-graph theory but with extensions. According to the two-graph theory, symbolic terms can be generated by enumerating spanning tree-pairs that span a pair of V-graph and I-graph while satisfying certain conditions. Each pair of admissible trees determines a product term. The creation of V-graph and I-graph is quite intuitive, but a set of formal rules should be satisfied, which are stated in [17]. Simply speaking, resistive/conductive edges like G_{oi} ,

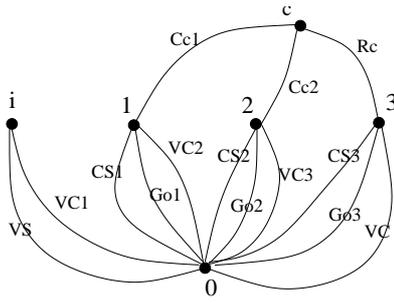


Fig. 6. Graph representation of the multi-stage schematic.

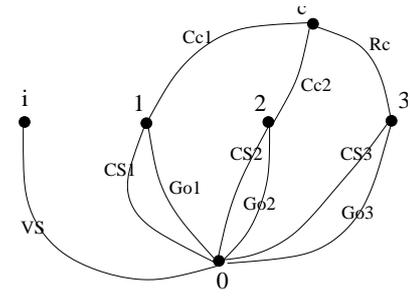


Fig. 8. I-graph of the multi-stage schematic.

C_{ci} , and R_c should appear in both V-graph and I-graph; VC-CS edge pairs are always distributed across the two graphs, with VC in the V-graph and CS in the I-graph. So does the VC-VS pair, but with the exception on the VS and CC edges. GPDD theory stipulates that all VS and CC edges be included in both V-graph and I-graph and, moreover, each admissible spanning tree-pair must include all VS and CC edges. The last rule is special in the GPDD theory (the classical two-graph theory does not subsume such cases). The work by Wambach et al. [23] was based on the two-graph theory as well, but it neither covered circuit elements involving CC and VS edges nor did it adopt BDD in tree-pair enumeration.

Following the preceding preparation, the graph in Fig. 6 can be broken into a pair of subgraphs shown in Figs. 7 and 8. These two subgraphs are to be used as the beginning pair of graphs for admissible tree-pair enumeration. It is important to note that GPDD does not generate terms by explicit tree-pair enumeration, rather it does that by a *graph reduction* process, in which a shared BDD is constructed. This makes up the key novelty of the GPDD construction method in the sense that it not only improves the enumeration efficiency substantially, but also greatly eases the implementation of many post-processing tasks (see [34]).

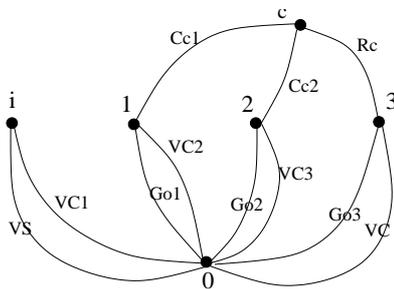


Fig. 7. V-graph of the multi-stage schematic.

Symbolic terms generated by GPDD correspond to all admissible spanning tree-pairs of the pair of V-graph and I-graph. An admissible spanning tree-pair means a pair of spanning trees, one spanning the V-graph while the other spanning the I-graph. Except for those common edges, cross-paired edges must correspond to a dependent element. VS and CC edges are those special edges that can appear either as a common edge or as a cross-paired edge. Spanning trees satisfying these conditions boil down to a finite set of terms

that form the symbolic transfer function of the given circuit.

When our goal is to extract poles and zeros, we are going to establish further rules that can differentiate significant terms from those insignificant. To that end, we shall further inspect the token counts among the the GPDD-generated terms, which turn out to be a meaningful metric for term differentiation.

Let us take a look at the term representing the dc gain, which is the coefficient b_0 in (2):

$$A_{dc} = b_0 = R_{o1}R_{o2}R_{o3}\mathbf{G}_{m1}\mathbf{G}_{m2}\mathbf{G}_{m3} \quad (6)$$

where the second term $R_{o3}R_{o1}\mathbf{G}_{m1}\mathbf{G}_{mf}$ in the expression (2) has been dropped due to the assumption $G_{mf} = 0$. Term b_0 results from the normalization, i.e., B_0/A_0 , where B_0 is the constant term of $N(s)$ and A_0 is the constant term of $D(s)$. GPDD produces that

$$B_0 = -G_{m1}G_{m2}G_{m3}R_c^{-1} \quad (7)$$

$$A_0 = R_{o1}^{-1}R_{o2}^{-1}R_{o3}^{-1}R_c^{-1} \quad (8)$$

The drawing of the tree-pairs corresponding to the above two terms is omitted. Note that when considering the dc term, the output admittances are simplified to $G_{oi} = R_{oi}^{-1}$ for $i = 1, 2, 3$, with the C-elements ignored. We note that GPDD always generates the impedance tokens in all terms in the reciprocal form, i.e., admittances (see [17]).

We see that the normalized dc term b_0 exhibits the gain factors $R_{o1}\mathbf{G}_{m1} \times R_{o2}\mathbf{G}_{m2} \times R_{o3}\mathbf{G}_{m3}$ explicitly. It is clear that a term including more number of gain factors would be more dominating than other terms with less number of gain factors. This principle is going to serve as the rule for term selection. As a matter of fact, similar rules are used by designers in manual analysis.

For multi-stage opamps the denominator constant term is always equal to the product of all resistors in reciprocal form, as can be verified by the GPDD theory. We called the normalization of all terms by A_0 *resistor-normalization* (or simply *R-normalization*).

The rest of the terms listed in the coefficients given in (2) and (3) can be verified by drawing the corresponding tree-pairs. We have implemented a general-purpose GPDD tool to generate all such terms. By listing the detailed terms, we would like to raise the reader's attention that the element tokens exhibiting in the terms are results of edge combinations in the admissible spanning tree-pairs. Not all symbol combinations may appear in the admissible terms. This fact leads to the following simple observation:

Observation 1: If certain circuit elements have been fixed in a term, then the rest of tokens in this term must not violate the spanning tree condition of GPDD.

This observation will be used as a baseline when we select terms that satisfy certain quantitative conditions. It can be detailed for the current example we are considering. The points listed below are going to serve as the basic hints for establishing term selection criteria.

- 1) If a term has a factor s^k , i.e., the k th degree of s , then this term must involve k C-elements. With any k C-elements fixed in the spanning tree-pair, the rest of circuit elements appearing in the term must not violate the admissible tree condition.
- 2) Because all nodes 1, 2, and 3 in the given circuit in Fig. 5 have a dc path to the ground by R_{oi} , they form a dc term A_0 (given in (8)) whose spanning tree pair consists of all three R_{oi} edges, the VS edge connecting node i to ground, and the edge of R_c .
- 3) The dc term A_0 defines the *dimension* of all terms generated by GPDD. After all terms are normalized by A_0 , then all terms become *dimensionless*.

We point out that the *term dimension* is very useful for measuring the token distribution in a symbolic term. For example, if a term is dimensionless, then the tokens in the term must have a balanced dimension. In other words, it means that the count of R's must be equal to the count of G_m 's and C's in one term. This fact can be used to make useful inferences for dimensionless terms like those listed in the following:

- If a normalized terms has k C's, p G_m 's, and q R's, then it must hold that $k + p = q$ because of dimension balancing.
- If we are interested in the compensation resistors R_c 's, we may further split the counts of R's into q_1 for regular R's and q_2 for the compensation Rc's, with $q_1 + q_2 = q$.
- Among several normalized terms with equal number of C's (which belong to the same coefficient set of s^k), if one term has an additional Rc comparing to another, this term must have an additional G_m so as to balance the dimension.

The above detailed token counts using the *dimension balancing property* turn out to be highly useful for measuring the term importance. We are going to establish term selection criteria based on the token counts. In the subsequent presentation, by a *coefficient set* we refer to the terms of equal s -degree in an s -expanded polynomial.

C. Term Selection Criterion

The term selection criteria are established in this section as a consequence of the preceding discussions. These criteria are based on the token counts in each term, hence can be implemented easily based on the existing GPDD software package. We have implemented it as an addon to the GPDD post-processing engine.

By definition, poles and zeros are polynomial roots. When GPDD generates product terms, the number of capacitors included in an admissible spanning tree-pair determines the s -order of the term. By collecting the same order of s -factored terms, two s -factorized polynomials can be obtained and saved

respectively at the two child vertexes of the GPDD root. This is done as a post-processing task. The two s -polynomials so produced become the numerator polynomial $N(s)$ and the denominator polynomial $D(s)$. A transfer function with all terms s -factored is called an s -expanded transfer function (or a rational function). Analytical poles and zeros can be derived from the two s -expanded polynomials by proper simplification. However, full symbolic polynomials directly generated by GPDD typically consist of a large set of symbolic terms. Poles and zeros expressed by such lengthy terms would not provide helpful insight for design, because such expressions are not readable or interpretable. Hence, term pruning is mandatory for generating interpretable pole/zero expressions. Here by *simplification* we just mean to drop those terms with low significance. In the procedure we describe below, simplification is carried out by working on each coefficient set. We do not work on reducing the polynomial order. Hence, the adopted simplification scheme is essentially different from those procedures adopted by model order reduction.

GPDD is a storage engine by saving terms as the paths in the diagram, where the symbols associated with the vertexes along each path are the term tokens. If one symbol is a capacitor, its token is Cs . If k capacitor symbols are involved in a path, this path would generate a term of order- k .

Two methods can be used to generate s -expanded polynomials. One is by generating terms one by one explicitly, counting the number of C's, then collecting the terms to produce two s -expanded polynomials $N(s)$ and $D(s)$. The other method is a recursive procedure that performs s -factorization by running a bottom-up traversal within a GPDD. The price is that we have to introduce an array in each GPDD vertex to save a record of the partially s -factorized coefficients. The result of recursion is a multi-root GPDD as illustrated in Fig. 9, each root provides one coefficient set.

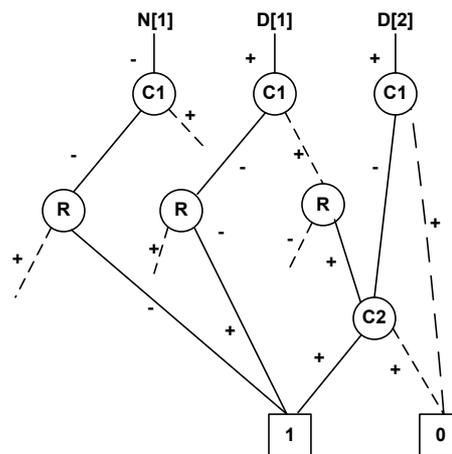


Fig. 9. Illustration of a multi-root GPDD.

Because the generated s -factored terms will be used for further processing with a variety of purposes, saving the terms in shared form in GPDD (as done by the second method) is more cost-efficient. Processing terms by going through the terms one by one (like counting tokens or running numerical

evaluation), as done by the first method, is not the most cost-efficient when the number of terms is large. Hence, in practice we would recommend the second method of post-processing, i.e., by using intermediate storage to generate a *multi-root* GPDD. Nevertheless, the detailed implementation can take a variety of forms.

The s -expanded polynomial coefficients are the base for deriving approximate symbolic poles/zeros. Since not all terms are of equal importance for generating approximate expressions, we need term selection criteria to compare the relative term significance. The proposed term selection scheme is based on comparing terms belonging to the same *coefficient set*.

In order to compare the terms in a coefficient set, we introduce a formal procedure based on the token counts. It turns out that the simple token counts are reminiscent of certain kind of design knowledge that designers commonly deal with. We find that human perception of *gain* and *compensation* can be quantitatively measured by certain token patterns and their counts. In this sense certain form of *design knowledge* can be quantitatively measured by introducing proper token counts. For example, Gain is symbolized by the product of G_m and a load resistance R that reflects the voltage amplification of one opamp stage. On the other hand, compensation consisting of capacitors, resistors, and G_m elements is purposely introduced by designers in feedback or feedforward paths. Different compensation elements play different roles in circuit operation. Compensation capacitors are used mainly to shape the frequency response so that the parasitic effects can be reduced, compensation resistors are used mainly for regulating the zeros, and the feedforward G_m 's offer the possibility of cancelling other current paths to regulate the zeros as well.

In analog design practice compensation capacitors typically dominate over the parasitic capacitors C_{oi} (often one magnitude of order larger [6]). Notwithstanding such a fact, we do not particularly assume that the parasitic capacitors are always negligible for pole-zero extraction. As a matter of fact, parasitic capacitors are occasionally considered for in-depth pole-zero analysis, such as in the study of mirror poles. For this reason, we decide not to differentiate the capacitor tokens during term selection. Instead, the term selection rules will be focused on the G_m and R_c tokens.

Since the compensation resistors (denoted by R_c 's) play a different role than the load resistors (denoted by R_o 's), we have to differentiate these two types of tokens during term selection. For a progressive presentation, we temporarily ignore all compensation resistors by assuming that all $R_c = 0$. This assumption simplifies the justification on term dominance argument. The first term selection criterion is stated below.

Rule 1: If all terms in a coefficient set do not involve R_c , then those terms with the maximum number of G_m 's are selected.

This rule implies the following: 1) If a coefficient set has only one term, this term must be selected. 2) If all terms of a coefficient set do not involve any G_m token, then all these terms are selected. Hence, this rule is effective on a coefficient set containing terms with different G_m counts.

The dimension balancing property can be used to justify

this rule. Because resistor-normalization renders all terms dimensionless, each G_m token must be balanced by an R token in each normalized term. Each $G_m R$ factor in a term enhances the significance of the term by amplification. The more such factors are involved in a term, the more dominant the term becomes. This is a typical guideline used by designers in selecting dominant terms. Hence, we say that a term selection rule incorporating such an ingredient partly reflects human *design knowledge*.

Next, we extend the term selection rule to include compensation R_c 's. Since R_c elements are introduced mainly for zero relocation, if terms with R_c tokens are selected, they must have comparable importance to those selected dominant terms without R_c . This observation leads us to the next term selection criterion. To facilitate statement, we denote by $\#R_c$ the count of R_c tokens in a term (similarly for G_m) and let $\#G_{m,\max} := \max\{\#G_m : \#R_c = 0\}$, where \max is taken over a coefficient set.

Rule 2: Terms of a coefficient set satisfying the following count inequality

$$\#G_m \geq \#G_{m,\max} + \#R_c \quad (9)$$

are selected.

Since $\#G_{m,\max}$ denotes the maximum count of G_m 's among those terms not having any R_c token, if a term contains an R_c token, then it is selected if and only if it contains *at least* $(\#G_{m,\max} + 1)$ G_m tokens. If two R_c tokens appear in a term, then this term (if selected) must have at least two extra G_m tokens to balance the two R_c tokens, and so on. The interpretation of this rule once again follows the intuition in design practice. Rule 2 basically requires that terms with R_c (called *Rc-terms*) are selected simply because they have significance comparable to those selected *non-Rc-terms* so that term cancellation can be performed. It is evident that Rule 1 is a special case of Rule 2. But we think that the statement of Rule 1 is conducive to the comprehension of Rule 2.

However, there exists another case that at least two terms exist in a coefficient set and all of them involve R_c -tokens, but none of them include G_m -tokens. This is possible because C-tokens can balance the dimension of those R_c -tokens. Such terms would violate the criterion (9), hence all terms would disappear if we do not modify Rule 2. To create a complete rule set, we decide to keep all terms in such a coefficient set and leave the decision to the designer to further prune insignificant terms. Therefore, we append the following rule for completeness.

Rule 3: If $\#R_c > 0$ but $\#G_m < \#R_c$ for all terms in a coefficient set, then all terms in that coefficient set are selected.

D. Comments on Implementation

The proposed term selection rules are based on the token counts only. Such rules can be implemented by post-processing the terms generated in a GPDD. We would like to comment on the specialty involved with the compensation R_c elements. In a circuit schematic or netlist we normally do not differentiate the output resistors from the compensation resistors. However, pole-zero oriented term selection requires us to make specific

counts on the R_c elements. Hence, we need a mechanism to annotate the R_c elements in a circuit. There could be several possible implementations. For example, we may annotate the R_c 's in a netlist if symbolic analysis starts from reading a netlist. This implementation would require parser support. If the circuit input is by drawing schematic on a graphical user interface (GUI), we may implement a schematic editor with device dialog, allowing the user to annotate the compensation resistors before performing symbolic pole-zero analysis.

Furthermore, we point out that counting tokens can be implemented with great ease based on the GPDD engine. It is very similar to traversing all terms (i.e., paths) in a GPDD. The complexity is no more than a constant multiple of the GPDD size (measured by the GPDD vertex counts, see [34]).

Regarding the runtime, because the macro-level opamp circuits are mostly small and of low order, generating a full transfer function by GPDD takes virtually inappreciable time (less than one second). On the other hand, rule-based term selection goes through all terms in a GPDD only once, it also takes no appreciable time. Hence, from reading a circuit description to generation of a simplified transfer function, including rough algebraic arrangement like common factor extraction, etc., the total runtime was hardly perceivable on a desktop computer in our test. The tool response is virtually instantaneous, hence such a tool can greatly benefit design exploration.

V. APPLICATIONS

This paper was largely motivated by the paper by Leung and Mok [2], where ten multi-stage opamps were studied. These opamps encompass a variety of compensation structures, covering simple Miller compensation two-stage amplifiers and three-stage amplifiers with sophisticated feedback and feedforward frequency compensations. Systematic design methodology has been presented in [2]. However, it seems that the authors conducted all pole-zero analysis manually.

Following the work [2], interest in the multi-stage opamp design has been surging, as evidenced by a long list of publications along the line [6], [7], [8], [9], [4], [10], [11], [5], [12], [13], [14], [16]. Slight inspection on the papers reveals that hardly any authors were using any kind of automatic pole-zero analysis tool to derive the simplified pole-zero expressions. Virtually all works presented simplified transfer function without carrying full explanation. It seems that deriving such simplified transfer functions is supposed to be basic technical knowhow with all design experts.

The main purpose of this section is to demonstrate that those pole-zero oriented transfer function simplification results reported in the literature can be automatically generated by using the proposed symbolic GPDD tool enhanced with the term selection strategies. To facilitate the experiment, we have extended our in-house AICE (Analog IC Explorer) software by adding the pole-zero generation functionality. The tool was written in C++, furnished with a graphical user interface, so that user may input a circuit by drawing a schematic in the schematic editor. Users may click a device dialog to specify device attributes like annotating a resistor to be a compensation

R_c , etc. The auto-generated poles and zeros are displayed in a well-organized HTML format.

We shall use the circuits and the simplified transfer functions reported in the cited papers as the verifiers. We have included a total of 11 circuits in the following test report. The reader is reminded that by generating a simplified low-order symbolic transfer function, the symbolic poles and zeros are considered generated because typically designers would apply the *root-splitting technique* to derive the pole-zero expressions from the simplified polynomial coefficients (please refer to those cited papers).

A. Test Report

Circuit 1. The two-stage SMC (simple Miller compensation) opamp is shown in Fig. 10. AICE generates the approximate transfer function (TF) given in Eqn. (10) (in the appendix). In the original exact TF the first-order coefficient of $D(s)$ has five terms, four of them are pruned. This is the only part to which simplification is applied. Usually designer would assume that C_{o1} is ignorable. We would leave such decisions to the designer.

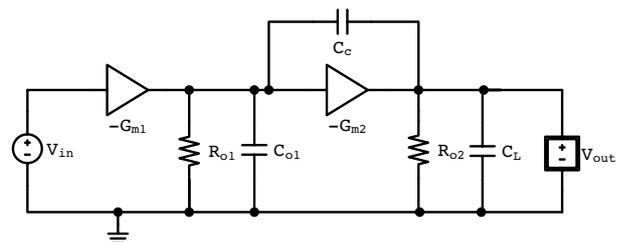


Fig. 10. Circuit 1: Two-stage SMC opamp [2], whose generated TF is given in (10).

Circuit 2. The two-stage SMCNR (simple Miller compensation with nulling resistor) opamp is shown in Fig. 11. AICE generates the approximate TF given in Eqn. (11) (in the appendix). In the original exact TF the first-order coefficient of $D(s)$ contains six terms, five of them are pruned. Two terms with R_c in the second-order coefficient of $D(s)$ are dropped. Comparing to equation (7) in [2] we find that the authors of [2] included terms with R_c in the denominator. But due to the choice of $R_c = 1/G_{m2}$ for eliminating the zero, such terms with R_c are actually relatively small, therefore can be dropped. This test validates that the term selection criterion (9) works quite well.

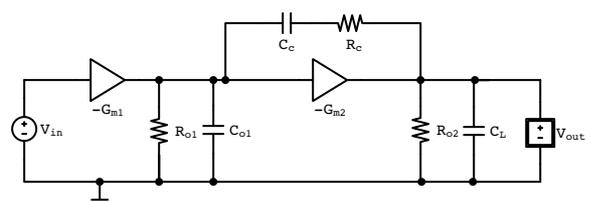


Fig. 11. Circuit 2: Two-stage SMCNR opamp [2], whose generated TF is given in (11).

Circuit 3. This is a two-stage opamp using compensation by multi-path zero cancellation (MZC) [2]. The schematic is shown in Fig. 12 and the simplified TF is given in Eqn. (12). This circuit is relatively simple. We just use it to verify the correctness of the proposed term selection criteria.

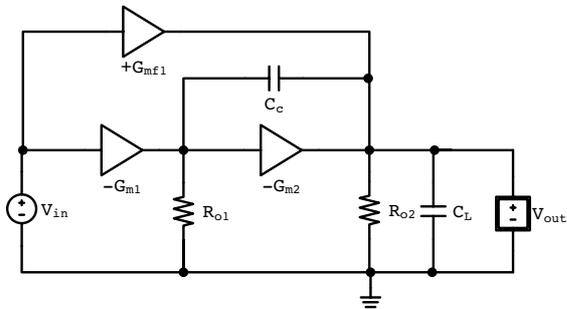


Fig. 12. Circuit 3: Two-stage MZC opamp [2], whose generated TF is given in (12).

Circuit 4. Three-stage opamps often apply nested Miller compensation (NMC). Fig. 13 shows the schematic of such a compensation structure without using any nulling resistor. Eqn. (13) gives the simplified TF. The derived TF matches exactly Eqn. (10) given in the paper [2].

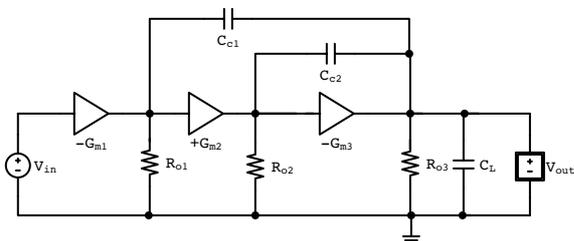


Fig. 13. Circuit 4: Three-stage NMC opamp [2], whose generated TF is given in (13).

Circuit 5. Shown in Fig. 14 is a multi-path NMC (MNMC) compensated three-stage opamp. Its simplified TF is derived in Eqn. (14). Comparing to the counterpart in [2], we see that Eqn. (23) derived by Leung and Mok is simpler because they applied further simplification under the assumption that $G_{m3} \gg G_{m1}, G_{m2}$. By this assumption the second-order terms in the numerator of the auto-generated equation (14) can be ignored and the second term of the first-order in the numerator can be dropped as well, resulting in the further simplified expression in Leung and Mok's paper. This example hints that, depending on the designer's needs, the machine-generated results can be further pruned by designer.

Circuit 6. Shown in Fig. 15 is a three-stage opamp with nested G_m -C compensation. Its simplified TF is derived in Eqn. (15). This result is exactly identical to Eqn. (29) derived in [2]. With the explicit coefficient expressions available, designer can quickly perceive that choosing appropriate feedforward gain factors can eliminate the zeros. For example, the choice of $G_{mf1} = G_{m1}$ and $G_{mf2} = G_{m2}$ was made in [2].

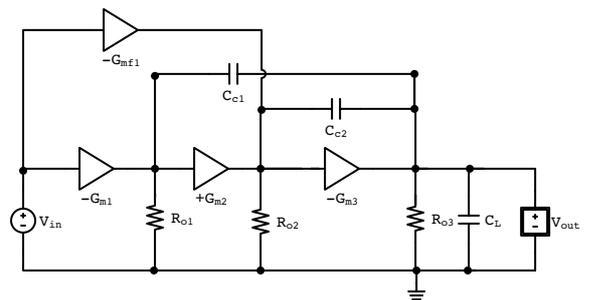


Fig. 14. Circuit 5: Three-stage MNMC opamp [2], whose generated TF is given in (14).

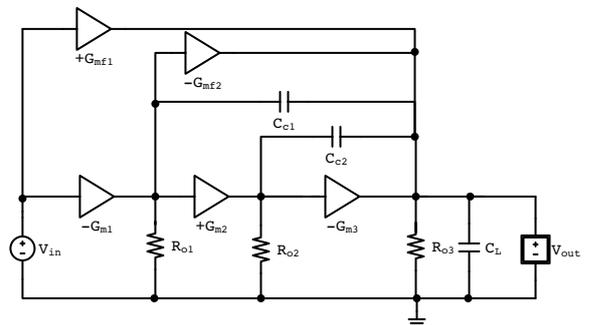


Fig. 15. Circuit 6: Three-stage NGCC opamp [2], whose generated TF is given in (15).

Circuit 7. The three-stage opamp with the compensation structure shown in Fig. 16 was originally proposed by Leung and Mok [2]. The compensation structure is called *damping-factor-control frequency compensation* (DFCFC). Two structures with the similar compensation were considered in that work. The schematic shown in Fig. 16 was referred to as DFCFC1 in [2]. The simplified TF is derived in Eqn. (16). This function looks lengthy. But interestingly, when we look at the poles and zeros generated automatically by the root-splitting principle (i.e., by dividing the successive polynomial coefficients, we find that the first zero and the second pole are exactly equal, i.e.,

$$z_1 = p_2 = -\frac{G_{m2}G_{m3}}{R_{o4}C_{c2}(G_{m4}G_{mf2} + G_{m2}G_{m3})}$$

After cancelling this pair of pole-zero, we get the TF given by Eqn. (37) in the paper [2].

Circuit 8. So far we have studied three-stage opamps with capacitive feedback compensation and feedforward G_m compensation. We have not considered the nulling resistor compensation. The selection criterion (9) has to be tested by circuits with nulling resistors. The circuit shown in Fig. 1 is one of such circuits. References [2], [6] studied this circuit by assuming the zero feedforward gain, while reference [10] considered a nonzero G_{mf} . The simplified TF is given in (17), which is identical to that derived in [2], [6] without G_{mf} .

Circuit 9. In this example we consider the most general NMCNR structure with three general nulling resistors shown in Fig. 17. Two special cases of this compensation structure

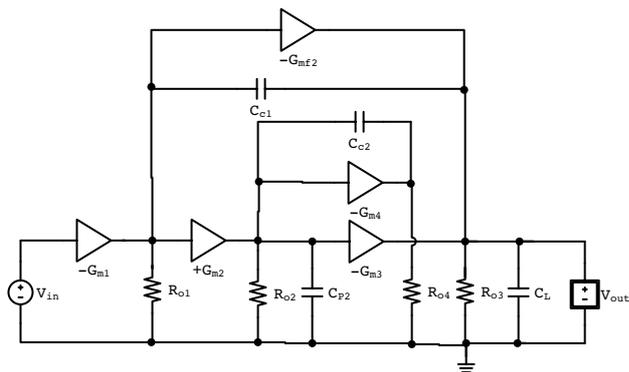


Fig. 16. Circuit 7: Three-stage opamp named DFCFC1 [2], whose generated TF is given in (16).

were considered in [6], with the TFs derived in Eqns. (12) and (14) there. The first case corresponds to the circuit with nonzero R_{c1} , R_{c2} , and zero R_c while the second case corresponds to the circuit with $R_{c1} = R_{c2} = 0$ but nonzero R_c . The AICE tool generated the result in Eqn. (18), which is quite lengthy, but has included both results in the paper [6] as the special cases. This test demonstrates that the AICE tool can be used either for general compensation structures or for certain simplified compensation structures.

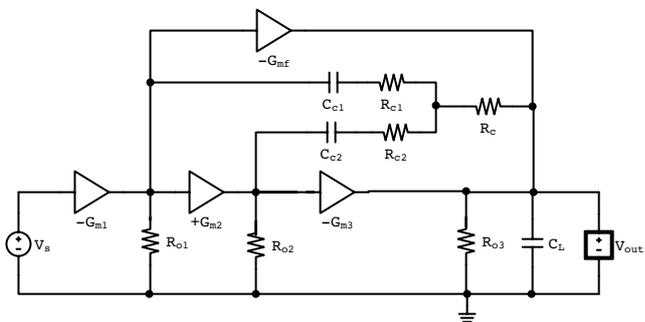


Fig. 17. Circuit 9: Three-stage NMCNR opamp with three nulling resistors [6], whose generated TF is given in (18).

Circuit 10. The three-stage opamp shown in Fig. 18 was studied in the paper [11] by Guo and Lee. The compensation structure was called *single-capacitor active-feedback compensation*, whose application target was toward small-capacitive load, for instance, internal amplifiers used in analog-to-digital converters. The proposed feedback compensation contains a current buffer which breaks the feedforward high-frequency current path. It is able to eliminate the dominant right-half plane zero.

The simplified TF is derived in Eqn. (19). Comparing to Eqn. (1) provided in [11], the only discrepancy is in the third-order terms of the denominator $D(s)$. Since this circuit does not employ any nulling resistor, all five terms in the third-order coefficient satisfy the second term selection criterion. Because the authors of [11] assumed that $R_1, R_2, R_3 \gg R_a$, they reserved only one of the five terms in their paper. We have underlined those terms in (19), which were dropped in [11].

This example once again testified a basic principle that a designer should be aware of using the AICE tool. When the symbolic tool generates more terms than a designer might need, the designer has the full authority to further prune the terms based on his/her design considerations. A successful tool should not omit those significant terms. Hence, although simplified, the auto-generated symbolic terms should still guarantee to be a superset of all terms most relevant to design.

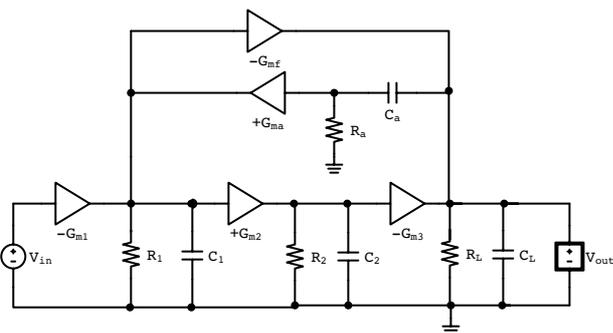


Fig. 18. Circuit 10: Three-stage SCAFC opamp [11], whose generated TF is given in (19).

Circuit 11. The three-stage compensation scheme proposed in paper [7] belongs to the category of *reversed nested Miller compensation (RNMC)*. The proposed compensation includes a voltage buffer and a nulling resistor. This circuit is abbreviated as *RNMCVB (reversed NMC with voltage buffer)*. The circuit schematic is shown in Fig. 19. The voltage buffer is assumed to drive an output resistance R_{bo} . In the AICE schematic the voltage buffer is described by a VCVS element. After symbolic generation, the symbol E can be substituted by the value $E = 1$.

The auto-generated transfer function is shown in Eqn. (20). Comparing it to the exact symbolic result, we see that the numerator $N(s)$ in (20) is exact, i.e., all terms have been preserved. In the denominator, quite a number of terms (actually 10 terms) have been eliminated from the second-order coefficient. Comparing to Eqn. (1) derived in [7], the only difference is that the term with R_c (underlined) in the third-order coefficient of the denominator is dropped in the paper [7]. This is because designers often assume that R_{o1} dominates R_c .

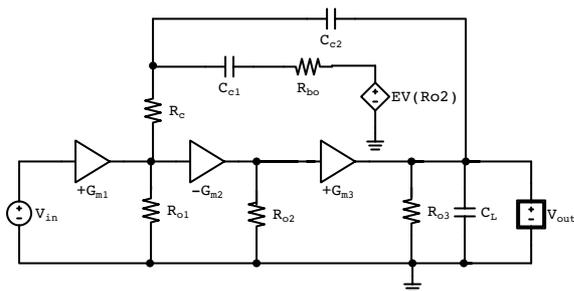


Fig. 19. Circuit 11: Three-stage RNMCVB opamp [7], whose generated TF is given in (20).

B. Numerical Accuracy

We use one example, i.e., *Circuit 8* given in Fig. 1 with $G_{mf} = 0$, to verify the numerical accuracy after symbolic PZ approximation. Given appropriate circuit element values, the comparison between the exact and the simplified transfer functions is plotted in Fig. 20. The approximation curve agrees with the accurate one quite well.

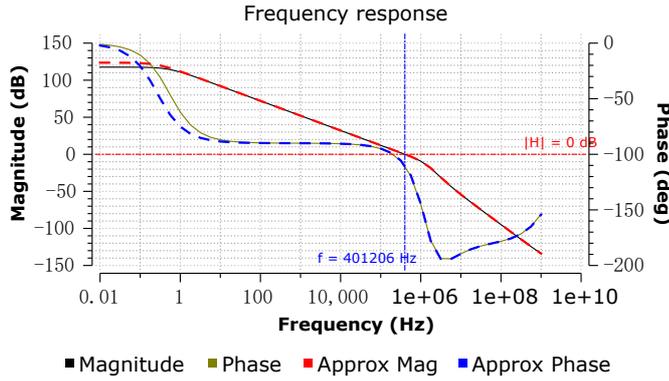


Fig. 20. Frequency response comparison for Circuit 8.

C. Comments on Comparison

The pole-zero analysis method developed in this work is targeted at multi-stage opamps involving frequency compensations. Our method is formal and purely rule based, it differentiates itself from those existing ones based on numerical reference. All numerical reference based methods do not explore the implication of symbolic tokens and their combinations. However, design practice reminds us that the symbolic tokens as they appear in a term actually embody quite an amount of the underlying circuit properties like gain factor or compensation. Numerical methods cannot fully explore such implications. Hence, comparing two categories of methods founded on two differing philosophies cannot be made fair. For example, different choices of the compensation element values can lead to biased term selection by a numerical method, while a token-count based criterion like (9)

is unbiased by the choice of element values. Therefore, we decide not to compare our method to those numerical reference based methods, because they might completely ignore the compensation terms, resulting in useless pole-zero expressions.

The comparison work in this paper has been targeted mainly at those manually derived pole-zero results reported in the existing circuit design papers. Matching the human-derived results is a fair justification on the merits of incorporating design knowledge in the term selection rules.

VI. CONCLUSION

The main contribution of this paper is the formulation of a topology-based framework for automatic pole-zero extraction in interpretable form. The developed term selection criteria turn out to be simple but effective. The reported extensive test experiment has justified that the proposed pole-zero analysis method can be successfully applied to a variety of multi-stage operational amplifiers with versatile compensation structures. It has been demonstrated that the proposed tool can finish one round of symbolic pole-zero analysis in inappreciable time for macro-level multi-stage opamps, providing interpretable simplified transfer functions that allow user to further manipulate the expressions without too much extra effort. With such a tool, designers can be liberated from routine manual derivation work, and can turn to dedicate more efforts to creative work like exploring new circuit configurations.

Recently, there is a surge of research interest in the subject on using multi-stage opamps that are capable of driving large capacitive loads, such as [14], [13], [16] among others. We believe that the proposed CAD tool can be used to explore more compensation possibilities, like using a single Miller compensation in multi-stage design. These are subjects for further research.

VII. ACKNOWLEDGEMENT

The author is grateful to the anonymous reviewers for their constructive comments.

APPENDIX

A. Generated Approximate Transfer Functions

$$\frac{V_{out}}{V_{in}} \approx \frac{R_{o1}R_{o2}G_{m1}G_{m2} - R_{o1}R_{o2}G_{m1}C_c s}{1 + R_{o1}R_{o2}G_{m2}C_c s + R_{o1}R_{o2}(C_L C_c + C_L C_{o1} + C_c C_{o1})s^2} \quad (10)$$

$$\frac{V_{out}}{V_{in}} \approx \frac{R_{o2}G_{m1}G_{m2}R_{o1} + R_{o2}G_{m1}C_c R_{o1}(R_c G_{m2} - 1)s}{1 + R_{o2}G_{m2}C_c R_{o1}s + R_{o2}R_{o1}(C_L C_c + C_L C_{o1} + C_c C_{o1})s^2 + R_c C_L R_{o2}C_c R_{o1}C_{o1}s^3} \quad (11)$$

$$\frac{V_{out}}{V_{in}} \approx \frac{R_{o1}G_{m2}G_{m1}R_{o2} + R_{o1}C_c R_{o2}(G_{mf1} - G_{m1})s}{1 + R_{o1}C_c G_{m2}R_{o2}s + R_{o1}C_c R_{o2}C_L s^2} \quad (12)$$

$$\frac{V_{out}}{V_{in}} \approx \frac{R_{o1}G_{m1}R_{o3}G_{m2}G_{m3}R_{o2} - R_{o1}G_{m1}R_{o3}G_{m2}C_c R_{o2}s - R_{o1}C_{c1}G_{m1}R_{o3}C_c R_{o2}s^2}{1 + R_{o1}C_{c1}R_{o3}G_{m2}G_{m3}R_{o2}s + R_{o1}C_{c1}R_{o3}C_c R_{o2}(-G_{m2} + G_{m3})s^2 + R_{o1}C_{c1}C_L R_{o3}C_c R_{o2}s^3} \quad (13)$$

$$\frac{V_{out}}{V_{in}} \approx \frac{R_{o1}G_{m1}R_{o3}G_{m2}G_{m3}R_{o2} + R_{o1}R_{o3}R_{o2}(C_{c1}G_{m3}G_{mf1} - G_{m1}G_{m2}C_{c2})s + R_{o1}C_{c1}R_{o3}C_{c2}R_{o2}(-G_{m1} - G_{mf1})s^2}{1 + R_{o1}C_{c1}R_{o3}G_{m2}G_{m3}R_{o2}s + R_{o1}C_{c1}R_{o3}C_{c2}R_{o2}(-G_{m2} + G_{m3})s^2 + R_{o1}C_{c1}C_L R_{o3}C_{c2}R_{o2}s^3} \quad (14)$$

$$\frac{V_{out}}{V_{in}} \approx \frac{R_{o1}G_{m1}R_{o3}G_{m2}G_{m3}R_{o2} + R_{o1}G_{m1}R_{o3}C_{c2}R_{o2}(G_{mf2} - G_{m2})s + R_{o1}C_{c1}R_{o3}C_{c2}R_{o2}(G_{mf1} - G_{m1})s^2}{1 + R_{o1}C_{c1}R_{o3}G_{m2}G_{m3}R_{o2}s + R_{o1}C_{c1}R_{o3}C_{c2}R_{o2}(-G_{m2} + G_{mf2} + G_{m3})s^2 + R_{o1}C_{c1}C_L R_{o3}C_{c2}R_{o2}s^3} \quad (15)$$

$$\frac{V_{out}}{V_{in}} \approx \frac{R_{o2}G_{m3}G_{m2}R_{o3}G_{m1}R_{o1} + R_{o4}R_{o2}C_{c2}R_{o3}G_{m1}R_{o1}(G_{m4}G_{mf2} + G_{m3}G_{m2})s + R_{o4}R_{o2}C_{c2}R_{o3}G_{m1}R_{o1}(G_{mf2}C_{P2} - G_{m4}C_{c1})s^2 - R_{o4}R_{o2}C_{P2}C_{c2}R_{o3}G_{m1}C_{c1}R_{o1}s^3}{1 + R_{o2}G_{m3}G_{m2}R_{o3}C_{c1}R_{o1}s + R_{o4}R_{o2}C_{c2}R_{o3}C_{c1}R_{o1}(G_{m4}G_{mf2} + G_{m3}G_{m2})s^2 + R_{o4}R_{o2}C_{c2}R_{o3}C_{c1}R_{o1}(G_{mf2}C_{P2} + G_{m4}C_L)s^3 + R_{o4}R_{o2}C_{P2}C_{c2}R_{o3}C_L C_{c1}R_{o1}s^4} \quad (16)$$

$$\frac{V_{out}}{V_{in}} \approx \frac{R_{o2}G_{m3}G_{m2}R_{o3}G_{m1}R_{o1} + R_{o2}G_{m2}R_{o3}G_{m1}R_{o1}(-C_{c2} + R_c C_{c2}G_{m3} + R_c G_{m3}C_{c1})s + R_{o2}C_{c2}R_{o3}G_{m1}C_{c1}R_{o1}(-1 + R_c G_{m3})s^2}{1 + R_{o2}G_{m3}G_{m2}R_{o3}C_{c1}R_{o1}s + R_{o2}C_{c2}R_{o3}C_{c1}R_{o1}(-G_{m2} + G_{m3})s^2 + R_{o2}C_{c2}R_{o3}C_L C_{c1}R_{o1}(-R_c G_{m2} + 1)s^3} \quad (17)$$

$$\frac{V_{out}}{V_{in}} \approx \frac{R_{o2}G_{m3}G_{m2}R_{o3}G_{m1}R_{o1} + R_{o2}R_{o3}G_{m1}R_{o1}(-C_{c2}G_{m2} + R_c G_{m3}G_{m2}C_{c1} + G_{mf}C_{c2} + G_{m3}G_{m2}C_{c1}R_{c1} + C_{c2}G_{m3}G_{m2}R_{c2} + R_c C_{c2}G_{m3}G_{m2})s + R_{o2}C_{c2}R_{o3}G_{m1}C_{c1}R_{o1}(-1 - G_{m2}R_{c1} + G_{mf}R_{c1} + G_{mf}R_c + G_{m3}G_{m2}R_{c1}R_{c2} + R_c G_{m3} + R_c G_{m3}G_{m2}R_{c2} + R_c G_{m3}G_{m2}R_{c1})s^2}{1 + R_{o2}G_{m3}G_{m2}R_{o3}C_{c1}R_{o1}s + R_{o2}C_{c2}R_{o3}C_{c1}R_{o1}(-G_{m2} + G_{mf} + G_{m3} + G_{m3}G_{m2}R_{c2})s^2 + R_{o2}C_{c2}R_{o3}C_L C_{c1}R_{o1}(-R_c G_{m2} + 1)s^3} \quad (18)$$

$$\frac{V_{out}}{V_{in}} \approx \frac{R_1G_{m1}R_LG_{m2}G_{m3}R_2 + R_1C_aG_{m1}R_LG_{m2}G_{m3}R_2R_a s + R_1C_aG_{m1}R_LR_2C_2R_aG_{mf}s^2}{1 + R_1C_aR_LR_2C_2R_aG_{ma}s + R_1C_aR_LR_2C_2R_aG_{ma}G_{mf}s^2 + (C_aC_LR_LR_2C_2R_a + R_1C_1C_aC_LR_LR_a + R_1C_1C_aR_2C_2R_a + R_1C_1C_aR_LR_2C_2 + R_1C_1C_LR_LR_2C_2)s^3 + R_1C_1C_aC_LR_LR_2C_2R_a s^4} \quad (19)$$

$$\frac{V_{out}}{V_{in}} \approx \frac{-R_{o2}G_{m3}G_{m2}R_{o3}G_{m1}R_{o1} + R_{o3}G_{m1}R_{o1}(C_{c2} - R_{o2}C_{c2}G_{m3}G_{m2}R_c - R_{o2}G_{m3}G_{m2}C_{c1}R_{bo} - R_{o2}G_{m3}G_{m2}C_{c1}R_c)s + C_{c2}R_{o3}G_{m1}C_{c1}R_{o1}(R_{bo} - R_{o2}G_{m3}G_{m2}R_cR_{bo} - R_{o2}G_{m2}R_cE)s^2}{1 + R_{o2}C_{c2}G_{m3}G_{m2}R_{o3}R_{o1}s + R_{o2}G_{m2}R_{o3}C_{c1}R_{o1}(C_{c2}G_{m3}R_{bo} + C_{c2}E + C_LE)s^2 + C_{c2}R_{o3}C_L C_{c1}R_{bo}(R_c + R_{o1})s^3} \quad (20)$$

REFERENCES

- [1] R. G. H. Eschauzier and J. H. Huijsing, *Frequency Compensation Techniques for Low-power Operational Amplifiers*. Dordrecht, The Netherlands: Kluwer Academic, 1995.
- [2] K. N. Leung and P. K. T. Mok, "Analysis of multistage amplifier – frequency compensation," *IEEE Trans. on Circuits and Systems – I: Fundamental Theory and Applications*, vol. 48, no. 9, pp. 1041–1056, September 2001.
- [3] G. Palumbo and S. Pennisi, *Feedback Amplifiers: Theory and Design*. Boston, MA: Kluwer, 2002.
- [4] A. D. Grasso, G. Palumbo, and S. Pennisi, "Analytical comparison of frequency compensation techniques in three-stage amplifiers," *International Journal of Circuit Theory and Applications*, vol. 36, no. 1, pp. 53–80, 2008.
- [5] A. D. Grasso, D. Marano, G. Palumbo, and S. Pennisi, "Analytical comparison of reversed nested Miller frequency compensation techniques," *International Journal of Circuit Theory and Applications*, vol. 38, no. 7, pp. 709–737, 2010.
- [6] G. Palumbo and S. Pennisi, "Design methodology and advances in nested-Miller compensation," *IEEE Trans. Circuits and Systems - I: Fundamental Theory and Applications*, vol. 49, no. 7, pp. 893–903, 2002.
- [7] K. P. Ho, C. F. Chan, C. S. Choy, and K. P. Pun, "Reversed nested Miller compensation with voltage buffer and nulling resistor," *IEEE Journal of Solid-State Circuits*, vol. SC-38, no. 10, pp. 1735–1738, Oct. 2003.
- [8] B. K. Thandri and J. Silva-Martinez, "A robust feedforward compensation scheme for multistage operational transconductance amplifiers with no Miller capacitors," *IEEE J. Solid-State Circuits*, vol. 38, no. 2, pp. 237–243, February 2003.
- [9] X. Fan, C. Mishra, and E. Sánchez-Sinencio, "Single Miller capacitor frequency compensation technique for low-power multistage amplifiers," *IEEE J. Solid-State Circuits*, vol. 40, no. 3, pp. 584–592, March 2005.
- [10] S. O. Cannizzaro, A. D. Grasso, R. Mita, G. Palumbo, and S. Pennisi, "Design procedures for three-stage CMOS OTAs with nested Miller-compensation," *IEEE Trans. on Circuits and Systems - I: Regular Papers*, vol. 54, no. 5, pp. 933–940, May 2007.
- [11] S. Guo and H. Lee, "Single-capacitor active-feedback compensation for small-capacitive-load three-stage amplifiers," *IEEE Trans. on Circuits and Systems-II: Express Briefs*, vol. 56, no. 10, pp. 758–762, October 2009.
- [12] Z. Yan, P.-I. Mak, M.-K. Lau, and R. P. Martins, "A 0.016-mm² 144-μW three-stage amplifier capable of driving 1-to-15 nF capacitive load with 0.95-MHz GBW," *IEEE J. Solid-State Circuits*, vol. 48, no. 2, pp. 527–540, February 2013.
- [13] A. D. Grasso, G. Palumbo, and S. Pennisi, "High-performance four-

stage CMOS OTA suitable for large capacitive loads," *IEEE Trans. on Circuits and Systems – I: Regular Papers*, vol. 62, no. 10, pp. 2476–2487, October 2015.

- [14] Z. Yan, W. Wang, P.-I. Mak, M.-K. Law, and R. P. Martins, "A 0.0045-mm² 32.4- μ W two-stage amplifier for pF-to-nF load using CM frequency compensation," *IEEE Trans. on Circuits and Systems – II: Express Briefs*, vol. 62, no. 3, pp. 246–250, March 2015.
- [15] Y. Xie and Z. Zhu, "A high speed four-stage operational amplifier in 65 nm CMOS," *Analog Integrated Circuits and Signal Processing*, vol. 86, pp. 133–140, 2016.
- [16] D. Marano, A. D. Grasso, G. Palumbo, and S. Pennisi, "Optimized active single-Miller capacitor compensation with inner half-feedforward stage for very high-load three-stage OTAs," *IEEE Trans. on Circuits and Systems – I: Regular Papers*, vol. 63, no. 9, pp. 1349–1359, September 2016.
- [17] G. Shi, "Graph-pair decision diagram construction for topological symbolic circuit analysis," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 2, pp. 275–288, February 2013.
- [18] G. E. Gielen, H. Walscharts, and W. Sansen, "ISAAC: A symbolic simulator for analog integrated circuits," *IEEE J. Solid-State Circuit*, vol. 24, no. 6, pp. 1587–1596, December 1989.
- [19] G. Gielen and W. Sansen, *Symbolic Analysis for Automated Design of Analog Integrated Circuits*. Norwell, MA: Kluwer Academic Publishers, 1991.
- [20] P. Wambacq, G. E. Gielen, and W. Sansen, "A cancellation-free algorithm for the symbolic simulation of large analog circuits," in *Proc. Int'l Symposium on Circuits and Systems*, 1992, pp. 1157–1160.
- [21] B. Razavi, *Design of Analog CMOS Integrated Circuits*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 2001.
- [22] P. R. Gray, P. J. Hurst, S. H. Lewis, and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, 5th ed. New York, NY, USA: John Wiley & Sons, Inc., 2009.
- [23] P. Wambacq, R. Fernández, G. E. Gielen, W. Sansen, and A. Rodríguez-Vázquez, "Efficient symbolic computation of approximated small-signal characteristics," *IEEE J. Solid-State Circuit*, vol. 30, no. 3, pp. 327–330, 1995.
- [24] G. Nebel, U. Kleine, and H. J. Pfeleiderer, "Symbolic pole/zero calculation using SANTAFE," *IEEE Journal of Solid-State Circuits*, vol. 30, no. 7, pp. 752–761, July 1995.
- [25] O. Guerra, J. D. Rodríguez-García, F. V. Fernández, and A. Rodríguez-Vázquez, "A symbolic pole/zero extraction methodology based on analysis of circuit time-constants," *Analog Integrated Circuits and Signal Processing*, vol. 31, pp. 101–118, 2002.
- [26] S. B. Haley, "The generalized eigenproblem: pole-zero computation," *Proceedings of the IEEE*, vol. 76, no. 2, pp. 103–120, 1988.
- [27] T. Dröge, G. Czysz and E. H. Horneber, "Symbolic pole and zero estimation for circuit design," in *Proc. of the 3rd IEEE Int. Conf. on Electronics, Circuits, and Systems (ICECS)*, August 1996, pp. 93–96.
- [28] E. Hennig, *Symbolic Approximation and Modeling Techniques for Analysis and Design of Analog Circuits*. Aachen, Germany: Shaker Verlag, 2000.
- [29] C. M. F. Constantinescu, M. Nitescu, "Computation of approximate symbolic pole/zero expressions," *Analog Integrated Circuits and Signal Processing*, vol. 40, no. 3, pp. 255–264, 2004.
- [30] J. Hsu and C. Sechen, "Accurate extraction of simplified symbolic pole/zero expressions for large analog ICs," in *Proc. IEEE Int. Symp. Circuits and Systems*, 1995, pp. 2083–2087.
- [31] W. Daems, G. Gielen, and W. Sansen, "Circuit simplification for the symbolic analysis of analog integrated circuits," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 4, pp. 395–406, April 2002.
- [32] H. Hu, G. Shi, A. Tai, and F. Lee, "Topological symbolic simplification for analog design," in *Proc. IEEE Int. Symp. on Circuits and Systems (ISCAS)*, Lisbon, Portugal, 2015, pp. 2644–2647.
- [33] G. Shi, H. Hu, and S. Deng, "Topological approach to automatic symbolic macromodel generation for analog ICs," *ACM Trans. on Design Automation of Electronic Systems*, 2016, accepted for publication.
- [34] G. Shi, S. X.-D. Tan, and E. Tlelo-Cuautle, *Advanced Symbolic Analysis for VLSI Systems – Methods and Applications*. New York: Springer, 2014.
- [35] R. E. Bryant, "Graph-based algorithms for boolean function manipulation," *IEEE Trans. on Computers*, vol. 35, no. 8, pp. 677–691, 1986.



Guoyong Shi (S'99-M'02-SM'11) received the B.S. in applied mathematics from Fudan University, Shanghai, China, the M.S. degree in electronics and information science from Kyoto Institute of Technology, Kyoto, Japan, and the Ph.D. degree in electrical engineering from Washington State University, Pullman, in 1987, 1997, and 2002, respectively.

From 2002 to 2005 he worked as a Post-Doctoral Research Scientist with the Department of Electrical Engineering, University of Washington, Seattle. He joined Shanghai Jiao Tong University in Shanghai, China, in 2006 and has been a Professor in Microelectronics since then. He has published more than 80 research papers in several technical areas. He is co-author of the book *Advanced Symbolic Analysis for VLSI Systems – Methods and Applications* published by Springer in 2014. His current research interests include design automation of analog/mixed-signal integrated circuits. He has served the ASPDAC technical program committee from 2013 to 2016. He is currently serving as an associate editor for *Integration*, the VLSI journal. Dr. Shi was co-recipient of the Donald O. Pederson Best Paper Award in 2007.