A GENERAL S-DOMAIN HIERARCHICAL NETWORK REDUCTION ALGORITHM

Sheldon X.-D. Tan

Department of Electrical Engineering
University of California, Riverside, CA 92521, USA
stan@ee.ucr.edu

ABSTRACT

This paper presents an efficient method to reduce complexities of a linear network in s-domain. The new method works on circuit matrices directly and reduces the circuit complexities by eliminating subcircuits in a hierarchical way. The resulting admittances in the reduced networks are kept as rational functions of s with reduced order. Some theoretical results are characterized for the presence of common factors coming from the suppression of subcircuits. A novel common factor removal (de-cancellation) strategy based on a graph-based hierarchical subcircuit reduction process is proposed. The resulting reduction algorithm is applicable to any linear circuits in s-domain. The stability of the reduced system is enforced by applying the Hurwitz polynomial approximation. The reduced systems can be used for fast s-domain analysis and for time domain waveform evaluation. Experimental results on both linear analog circuits and RLC circuits, and comparison with SPICE in s-domain analysis are also provided.

1. INTRODUCTION

As feature size keeps shrinking and clock rate continues roaring, physical effect related signal integrity issues become more severe[1]. Fast and accurate evaluation and modeling of analog/mixed-signal and interconnect circuits are critical for top-down mixed-signal SoC designs and interconnect-centered physical design and optimization [7, 8].

Due to the importance of many on-chip global interconnects like power /ground grid, global signal nets and clock trees, a number of projection-based model-order reduction based techniques have been introduced [4, 5, 6, 11, 12, 17, 18] to analyze the transient behavior of interconnects. Asymptotic Waveform Evaluation (AWE) algorithm was first proposed [12, 13] where explicit moment matching was used to compute the dominant poles at low frequencies. But AWE method is numerically unstable for higher order approximation. Thereafter a number of other projection-based model-order reduction methods based on implicit moment matching (via Krylov subspace projection) were developed. Examples are Pade via Lanczos (PVL) [4], Matrix PVL [5], Arnoldi method [17], Arnoldi Transformation method [18], PRIMA [11] and SyPVL algorithm [6]. The projection-based algorithms require the computations of the moments or Krylov subspace base vectors before reduction. So at least one DC solution of the whole circuit is required, which may not be efficient for highly connected structures like mesh-structured power distribution networks with millions of nodes. Also those projection-based methods are not efficient for circuits with many independent sources as independent sources can not be reduced by those methods in general.

Another approach to circuit complexity reduction is by means of local node reduction. The main idea is to reduce the number of nodes in the circuits and approximate the left elements in the circuit matrix in reduced rational forms. The major advantage of those methods over projection-based methods is that the reduction can be done in a local manner and no overall solutions of the whole circuit are required (with some realization techniques), which makes those methods very amenable to attack large linear networks. This idea has been explored by approximate Gaussian elimination for RC circuits [3], by direct truncation of the transfer function algorithm (DTT) [10] for tree-structured RLC circuits and an extended DTT method for non-tree structured RLC circuits [21]. Recently a topology based node-reduction method was proposed [14], in which nodes are reduced one at a time (topologically it is called Y-Δ transformation) and the generated admittance in the reduced network is represented as an order-reduced rational function of s. This method is equivalent to symbolic Gaussian elimination (s is the only symbol) but the reduction is done on circuit topologies only. The stability is enforced by Hurwitz polynomial approximation. But this method only works for linear circuits with limited element types (RCLK-VJ) and cannot be applied to reduce general linear circuits.

In this paper, we propose a new approach to reducing the complexities of linear circuits in s-domain. The new method performs the node reduction directly on the circuit matrices and hence it is general enough to reduce any linear circuits. Furthermore, instead of reducing one node at a time as done in [14]. The new method allows elimination of multiple nodes simultaneously. This leads to a general hierarchical s-domain analysis technique as we can suppress subcircuits, which consist of a number of nodes, one at a time in a hierarchical and an independent way. The new method thus is able to exploit the naturally hierarchical structures inherent in many linear circuits. Such a hierarchical node reduction is made possible by means of a graph based symbolic analysis technique for computing the new admittances and removing the common factors (de-cancellation) in determinants [15, 16] – the key operations in the new hierarchical node algorithm. Hurwitz polynomial approximation is also performed on the order-reduced transfer functions to enforce stability. The resulting algorithm can perform efficient s-domain analysis and time-domain analysis on any linear network with very high accuracy.

This paper is organized as follows. Section 2 reviews the matrix-based node reduction approach and concepts of DDDs and s-expanded DDDs for driving transfer functions of linear networks. Section 3 presents some theoretical results for the presence of common factors during subcircuit reduction processes. Section 4 presents our new approaches to the circuit complexity reduction on linear circuits. Experimental results are described in Section 5. Section 6 concludes the paper.

2. SUBCIRCUIT REDUCTION AND DDD GRAPHS

2.1. Subcircuit Reduction

In this section, we briefly review how nodes in a subcircuit can be suppressed in the matrix form. Consider a subcircuit with some internal structures and terminals. The circuit unknowns—the node-voltage variables and branch-current variables—can be partitioned into three disjoint groups $\mathbf{x}^I$, $\mathbf{x}^B$, and $\mathbf{x}^R$, where the superscripts *I, B, R* stand for, respectively, *internal* variables, *boundary* variables and the *rest* of variables. *Internal* variables are those local to the subcircuit, *boundary* variables (also called *tearing variables*) are those related to both the subcircuit and the rest of the circuit. Note that boundary variables include those variables required as the circuit inputs and outputs. With this, the system-equation set $\mathbf{Ax} = \mathbf{b}$, can be rewritten in the following form:

$$\begin{bmatrix} \mathbf{A}^{II} & \mathbf{A}^{IB} & 0 \\ \mathbf{A}^{BI} & \mathbf{A}^{BB} & \mathbf{A}^{BR} \\ 0 & \mathbf{A}^{RB} & \mathbf{A}^{RR} \end{bmatrix} \begin{bmatrix} \mathbf{x}^I \\ \mathbf{x}^B \\ \mathbf{x}^R \end{bmatrix} = \begin{bmatrix} \mathbf{B}^I \\ \mathbf{B}^B \\ \mathbf{B}^R \end{bmatrix}. \quad (1)$$

The matrix, $\mathbf{A}^{II}$, is the *internal* matrix associated with internal variable vector $\mathbf{x}^I$.

Subcircuit suppression is to eliminate all the variables in $\mathbf{x}^I$, and to transform (1) into the following reduced set of equations:

$$\begin{bmatrix} \mathbf{A}^{BB*} & \mathbf{A}^{BR} \\ \mathbf{A}^{RB} & \mathbf{A}^{RR} \end{bmatrix} \begin{bmatrix} \mathbf{x}^B \\ \mathbf{x}^R \end{bmatrix} = \begin{bmatrix} \mathbf{B}^{B*} \\ \mathbf{B}^R \end{bmatrix}, \quad (2)$$

where

$$\begin{aligned} \mathbf{A}^{BB*} &= \mathbf{A}^{BB} - \mathbf{A}^{BI}(\mathbf{A}^{II})^{-1}\mathbf{A}^{IB}, & (3) \\ &= \mathbf{A}^{BB} - \frac{1}{det(\mathbf{A}^{II})}\mathbf{A}^{BI}[\Delta^{II}_{u,v}]^T\mathbf{A}^{IB}, & (4) \end{aligned}$$

where $[\Delta_{u,v}]^T$ is called the *adjoint* matrix of $\mathbf{A}$, $\Delta_{u,v}$ is the first-order *cofactor* of $det(\mathbf{A})$ with respect to $a_{u,v}$, and matrix $\mathbf{A}_{a_{u,v}}$ is the $(n-1)\times(n-1)$-matrix obtained from matrix $\mathbf{A}$ by deleting row u and column v and

$$\begin{aligned} \mathbf{B}^{B*} &= \mathbf{B}^B - \mathbf{A}^{BI}(\mathbf{A}^{II})^{-1}\mathbf{B}^I, & (5) \\ &= \mathbf{B}^B - \frac{1}{det(\mathbf{A}^{II})}\mathbf{A}^{BI}[\Delta^{II}_{u,v}]^T\mathbf{B}^I. & (6) \end{aligned}$$

Subcircuit suppression can be performed for all the subcircuits by visiting the circuit hierarchy in a bottom-up fashion. Suppose that the number of internal variables is m, and the number of boundary variables is l. Eq.(4) and (6) can be written in the following expanded forms:

$$a^{BB*}_{u,v}(s) = a^{BB}_{u,v}(s) - \frac{1}{det(\mathbf{A}^{II}(s))}\sum_{k_1,k_2=1}^{m} a^{BI}_{u,k_1}(s)\Delta^{II}_{k_2,k_1}(s)a^{IB}_{k_2,v}(s), \quad (7)$$

where $u,v = 1,...,l$ and

$$b^{B*}_u(s) = b^B_u(s) - \frac{1}{det(\mathbf{A}^{II}(s))}\sum_{k_1,k_2=1}^{m} a^{BI}_{u,k_1}(s)\Delta^{II}_{k_2,k_1}(s)b^I_{k_2}(s), \quad (8)$$

where $u = 1,...,l$. From Equations (7) and (8), we can observe that admittance $a^{BB*}_{u,v}$ and input stimuli b^{B*}_u at boundary nodes will become rational functions of s once the subcircuit is suppressed.

In order to obtain the rational functions for $a^{BB*}_{u,v}(s)$ and $b^{B*}_u(s)$, we need to compute the rational function for a determinant whose elements may again be rational functions of s. This task can be achieved via determinant decision diagrams (DDDs), which was proposed originally for symbolic analysis of large analog circuits [15, 20]. We will show how the DDD graphs can be modified to compute the rational function for a determinant in the later sections.

An important issue is that we can only keep limited order of s for each rational function of s during the subcircuit reduction process. That implies that no common factors, which are functions of s, are allowed between the numerator and denominator of the rational functions in $a^{BB*}_{u,v}(s)$ and $b^{B*}_u(s)$. Otherwise, the reduced order is not correct. Such a *cancellation-free* rational functions bear the real order of the rational functions. Actually according to the definition of a determinant, it can be seen that the true (or cancellation-free) order of s in the denominators of both $a^{BB*}_{u,v}(s)$ and $b^{B*}_u(s)$ depends only on the product terms with the highest order of s in $det(\mathbf{A}^{II}(s))$, which in turn depends on the number of nodes reduced in the subcircuit, not on how those nodes are reduced. Notice also that the order of s changes from terms to terms, but the number of basic admittance from circuit devices in the denominators of the those newly generated terms are same for all the composite admittance terms. The number is called the *admittance order* of the term in the sequel.

But if the nodes in the subcircuit are reduced in different subcircuits at different hierarchical circuit-levels and are reduced at different time as shown later, common factors will automatically be generated in $a^{BB*}_{u,v}(s)$ and $b^{B*}_u(s)$. This has been observed already when nodes are reduced one at a time in $Y-\Delta$ transformation algorithm [14]. As a result, how to remove those common factors (de-cancellation) in the rational functions of $a^{BB*}_{u,v}(s)$ and $b^{B*}_u(s)$ in general becomes a key issue for the subcircuit reduction process. Fortunately, such cancellation-free rational functions can be obtained by efficient DDD graph operations. The DDD concept will be briefly reviewed in the following subsection.

2.2. Determinant Decision Diagrams

Determinant Decision Diagrams [15] are compact and canonical graph-based representation of determinants. A DDD is a signed, rooted, directed acyclic graph with two terminal vertices, namely the *0-terminal* vertex and the *1-terminal* vertex. Each non-terminal DDD vertex is labeled by a symbol in the determinant denoted by a_i, and a positive or negative sign denoted by $s(a_i)$. It *originates* two outgoing edges, called *1-edge* and *0-edge*. Each vertex a_i represents a symbolic expression $D(a_i)$ defined recursively as follows: $D(a_i) = a_i\ s(a_i)\ D_{a_i} + D_{\overline{a}_i}$, where D_{a_i} and $D_{\overline{a}_i}$ represent, respectively, the symbolic expressions of the vertices pointed by the 1-edge and 0-edge of a_i. The 1-terminal vertex represents expression 1, whereas the 0-terminal vertex represents expression 0. Such a DDD is called a complex DDD as each DDD node is function of complex frequency variable s.

To exploit the DDD to derive transfer rational functions of s, we need to directly represent circuit parameters not matrix entries. To this end, s-expanded DDDs are introduced [16]. The s-expanded DDD can be constructed from the complex DDD linearly in the size of the original complex DDD [16]. Once a coefficient DDD is constructed, its numerical value can be easily obtained by simply traversing all the vertices in the coefficient DDD graph and carrying out one addition and one multiplication at each vertex [15]. In this way, we can derive the rational function of s for any given determinant.

For our problem, some elements (in admittance forms) of a circuit matrix will become a rational function of s during the hierarchical subcircuit reduction process. As a result, the construction

of the rational function for such a determinant will be different from the method used for *s*-expanded DDDs [16]. A new DDD graph, called Y-expanded DDDs (YDDDs) are introduced for this task, which will be explained in Section 4.

3. CANCELLATION IN SUBCIRCUIT REDUCTION

In this section, we explain how cancellation happens from circuit devices and at different circuit levels and give some theoretical results.

3.1. Cancellation Due to Circuit Devices

The common factors will be introduced when the same circuit parameter will appear more than once in a circuit matrix in MNA formulation for subcircuits. Two cancellation patterns are shown in Fig. 1.

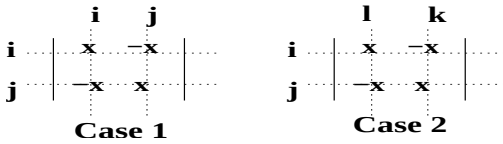

Figure 1: Matrix patterns causing term cancellation.

Case 1 comes from the rectangular appearance of a floating resistor in a admittance matrix and case 2 reflects the pattern from a voltage controlled current source (VCCS). As a result, all the product terms consisting of elements at (i,i) and at (j,j) will cancel all the terms consisting of elements at (i,j) and at (j,i) for case 1. The same is true for Case 2. Those canceling terms can cause two problems. First if the values of canceled terms are significantly larger than the actual values of some coefficients of rational functions, it will introduce large errors due to round-off errors. Second, notice that x and $-x$ are in the same row or column, they will be in different product terms and added together later. If x is a rational function (for example, we use $1/sL$ for inductance) a common factor among the numerator and the dominator will be introduced in the resulting rational function. For instance, let's consider addition of two rational functions $a_1/b_1 + a_2/b_2 = (a_1 * b_2 + a_2 * b_1)/(b_1 * b_2)$. If there is a common term c in both $b_1 = x*c$ and $b_2 = y*c$, c will become a common factor in both the numerator and the denominator of the resulting rational function.

Cancellation due to case 1 and case 2 in Fig. 1 can be easily removed during or after YDDD construction which is similar to *s*-expanded DDDs [16].

3.2. Cancellation at Middle Circuit Levels

Once a subcircuit has been suppressed, new cancellation patterns are created. This is illustrated in Fig.2

Nodes at (k_1,k_2) and at (t_1,t_2) are the internal nodes in a subcircuit to be suppressed. Nodes at (u_1,u_2) and at (v_1,v_2) are boundary nodes. Their connections with the two internal nodes are shown in the right-hand side of Fig. 2. As a result, admittances a_{u_1,k_2}, a_{k_1,u_2}, a_{v_1,t_2} and a_{t_1,v_2} are not zero. According to Eq.(7) we have,

$$a'_{v_1,u_2} = a_{v_1,u_2} - \frac{a_{v_1,t_2}\Delta^I_{k_1,t_2}a_{k_1,u_2}}{det(A^I)}, \quad (9)$$

$$a'_{u_1,v_2} = a_{u_1,v_2} - \frac{a_{u_1,k_2}\Delta^I_{t_1,k_2}a_{t_1,v_2}}{det(A^I)}, \quad (10)$$

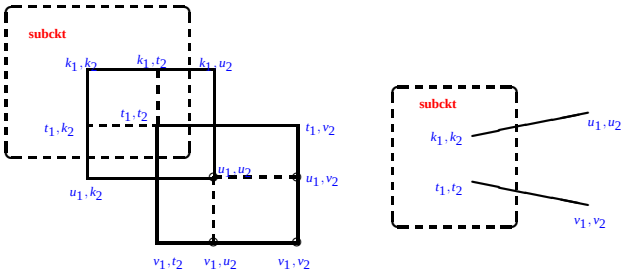

Figure 2: Cancellation patterns at middle circuit levels

$$a'_{u_1,u_2} = a_{u_1,u_2} - \frac{a_{u_1,k_2}\Delta^I_{k_1,k_2}a_{k_1,u_2}}{det(A^I)}, \quad (11)$$

$$a'_{v_1,v_2} = a_{v_1,v_2} - \frac{a_{v_1,t_2}\Delta^I_{t_1,t_2}a_{t_1,v_2}}{det(A^I)}, \quad (12)$$

where, $a'_{x,y}$ is the new admittance at (x,y). $det(A^I)$ is the internal system determinant of the subcircuit suppressed and $\Delta^I_{x,y}$ is the first-order cofactor of $det(A^I)$ with respect to element (x,y). $\frac{a_{v_1,t_2}\Delta^I_{k_1,t_2}a_{k_1,u_2}}{det(A^I)}$ terms is called *composite admittance* in the sequel. Since the four new composite admittances appear in two rows and two columns, they will cause new cancellations when node (u_1,u_2) and (v_1,v_2) are suppressed as the internal nodes in the upper level subcircuits. Specifically, we will have the following product terms in the internal system determinant of the upper-level subcircuit according to the definition of a determinant.

$$\begin{aligned} &a'_{u_1,u_2}a'_{v_1,v_2} - a'_{v_1,u_2}a'_{u_1,v_2} = \\ &R_x + \frac{a_{u_1,k_2}a_{v_1,t_2}\Delta^I_{k_1,k_2}\Delta^I_{t_1,t_2}a_{k_1,u_2}a_{t_1,v_2}}{det(A^I)det(A^I)} \\ &- \frac{a_{u_1,k_2}a_{v_1,t_2}\Delta^I_{k_1,t_2}\Delta^I_{t_1,k_2}a_{k_1,u_2}a_{t_1,v_2}}{det(A^I)det(A^I)} \end{aligned} \quad (13)$$

where R_x is the rest of other terms when the products of the new admittances are expanded. We have two scenarios to consider. First, if nodes (k_1,k_2) and (t_1,t_2) are the same node, i.e. $k_1 = t_1, k_2 = t_2$, the last two terms in (13) will cancel each other as $\Delta^I_{k_1,k_2}\Delta^I_{t_1,t_2} = \Delta^I_{k_1,t_2}\Delta^I_{t_1,k_2}$. Second, more generally if those two nodes are different internal nodes, we will show that the combined last two terms in (13) has a common factor in the numerator and the denominator. To see this, we first introduce the following theorem.

Theorem 1 *Given a matrix A and its four elements at rows i,j and columns m,n as shown in Fig. 3, we have*

$$\begin{vmatrix} \Delta_{im} & \Delta_{in} \\ \Delta_{jm} & \Delta_{jn} \end{vmatrix} = \Delta_{im}\Delta_{jn} - \Delta_{in}\Delta_{jm} = det(A)\Delta_{i,m,j,n}, \quad (14)$$

where $\Delta_{im,jn}$ is

$$\Delta_{im,jn} = (-1)^{i+m}(-1)^{j+n}det(A_{imjn}), \quad (15)$$

which is the second-order cofactor with rows i,j and columns m,n removed from $det(A)$.

The proof of the theorem is omitted due to limited space. Instead, we illustrate the theorem using the following 3×3 full matrix

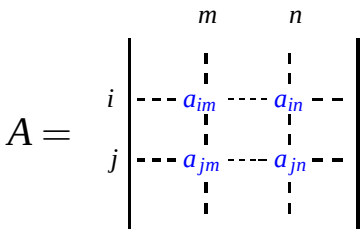

Figure 3: Illustration of Theorem 1

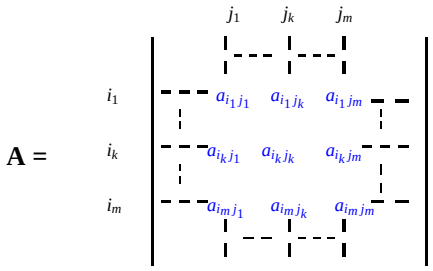

Figure 4: Illustration of Theorem 2

$A_{3\times 3}$.

$$A_{3\times 3} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Let's consider the following 4 first order cofactors with respect to a_{21}, a_{22}, a_{31}, and a_{32} respectively:

$$\Delta_{21} = -\begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} \quad \Delta_{22} = \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix}$$
$$\Delta_{31} = \begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix} \quad \Delta_{32} = -\begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix}$$

Then we have

$$\begin{aligned} & \Delta_{21}\Delta_{32} - \Delta_{22}\Delta_{31} \\ = \; & (a_{12}a_{11}a_{23}a_{33} - a_{12}a_{13}a_{21}a_{33} - a_{13}a_{11}a_{23}a_{32} + a_{13}^2 a_{32}a_{21}) \\ & -(a_{11}a_{12}a_{23}a_{33} - a_{11}a_{13}a_{22}a_{33} - a_{12}a_{23}a_{13}a_{31} + a_{13}^2 a_{31}a_{22}) \\ = \; & a_{13}^2(a_{32}a_{21} - a_{31}a_{22}) + a_{12}a_{13}(a_{23}a_{31} - a_{21}a_{33}) + \\ & a_{11}a_{13}(a_{22}a_{33} - a_{23}a_{32}) \\ = \; & a_{13}[a_{13}(a_{32}a_{21} - a_{31}a_{22}) + a_{12}(a_{23}a_{31} - a_{21}a_{33}) \\ & + a_{11}(a_{22}a_{33} - a_{23}a_{32})] \\ = \; & det(A_{3\times 3})\, a_{13} \\ = \; & det(A_{3\times 3})\, \Delta_{21,32} \end{aligned}$$

As a result, we know that $det(A^I)$ is a common factor in the combined last two terms in (13), which will become

$$\frac{a_{u_1,k_2} a_{v_1,t_2} \Delta^I_{k_1,k_2,t_1,t_2} a_{k_1,u_2} a_{t_1,v_2}}{det(A^I)} \tag{16}$$

after the common factor $det(A^I)$ in the numerator and the denominator cancels.

For any four composite admittances appearing at four rectangular positions (at two rows and columns), their corresponding Δ_{k_1,k_2} may have different position patterns in the original circuit matrix. If they appear also at four rectangular positions, the resulting product (*becoming higher order terms*) of those composite admittances will *merge* (after removal of common factors in numerators and denominators) as predicted by Theory 1. Otherwise (they only appear in only one row or one column or just one position) the two product terms will *cancel out* (become zero when added together) as $\Delta^I_{k_1,k_2}\Delta^I_{t_1,t_2}$ are identicial to $\Delta^I_{k_1,t_2}\Delta^I_{t_1,k_2}$ in Eq.(13).

In general, if the composite admittances from the suppression of one subcircuit populate more than two rows and columns, their corresponding first order cofactors Δ_{k_1,k_2} may also appear more than two rows and columns in the original circuit matrix. Detailed study shows if the number of rows or columns occupied by those first order cofactors are less than that of their composite admittances in the original circuit matrix, those higher-order admittance terms will eventually cancel out. But if the number of rows or columns of those first order cofactors equals to that of their corresponding composite admittances, we have the following more general theorem.

Theorem 2 *Given a matrix A and its $m \times m$ elements at rows $i_1, \ldots, i_m$ and columns $j_1, \ldots, j_m$ as shown in Fig. 4, we have*

$$\begin{vmatrix} \Delta_{i_1 j_1} & \ldots & \Delta_{i_1 j_k} & \ldots & \Delta_{i_1 j_m} \\ \Delta_{i_k j_1} & \ldots & \Delta_{i_k j_k} & \ldots & \Delta_{i_k j_m} \\ \Delta_{i_m j_1} & \ldots & \Delta_{i_m j_k} & \ldots & \Delta_{i_m j_m} \end{vmatrix} = det(A)^{m-1} \Delta_{i_1 j_1, \ldots, i_m, j_m}, \tag{17}$$

where $\Delta_{i_1 j_1, \ldots, i_m, j_m}$ is mth order cofactor with rows $i_1, \ldots, i_m$ and columns $j_1, \ldots, j_m$ removed from $det(A)$.

Notice that for a $n \times n$ matrix A, we have $A^{-1} = \frac{1}{det(A)}[\Delta_{u,v}]^T$, where $[\Delta_{u,v}]^T$ is the *adjoint* matrix of A. We have

$$\begin{aligned} det(A^{-1}) &= \frac{1}{det(A)^n} det([\Delta_{u,v}]^T]) \\ &= \frac{1}{det(A)^n} det([\Delta_{u,v}]] \end{aligned} \tag{18}$$

Given that $det(A^{-1}) \times det(A) = 1$, we have

$$det([\Delta_{u,v}]) = det(A)^{n-1} \tag{19}$$

This is the exact case in Theory 2 when m equals the matrix size n and $\Delta_{11,\ldots,nn} = 1$. In summary, we have the following result.

Corollary 1 *Let A^I be the internal system matrix of an immediate subcircuit, then $det(A^I)^{(k-1)}$ will be the common factor of the numerator and the denominator for all the terms consisting of k first-order cofactors of $det(A^I)$ when they are added together during the computation of the determinant consisting of those terms if they can't cancel out.*

As a result, we can see that any higher order terms coming from multiplications of composite admittances will eventually merge or cancel out with other same-order terms so that the resulting terms will have the correct admittance order. This is also consistent with the observation that no matter how internal nodes are reduced, the admittances in the reduced circuit should have the same admittance order given the same set of reduced nodes.

Also if the boundary nodes like (u_1, u_2) connect more than one internal nodes, the new admittances for the boundary nodes

will have more than one composite terms. But $det(A^I)^k$ will still be the common factor when those boundary nodes are suppressed at the next circuit level. To efficiently remove the common factors, we combine all the composite admittances with the same $det(A^I)$ in their denominator into one composite admittance. So the maximum number of merged composite admittances are equal to the number of immediate subcircuits suppressed at present circuit level at each element of circuit matrix. Each of the combined composite admittance will be represented by a YDDD node in the reduced circuit matrix. In this way, we can explicitly handle the cancellation situations via DDD-based graph operations to be shown in Section 4.

3.3. Cancellation at Different Circuit Levels

If the nodes (u_1,u_2) and (v_1,v_2) in Fig. 2 are reduced in different subcircuits at different circuit hierarchical levels, cancellation will happen when those composite admittance from different subcircuits are merged. To see this, suppose that only node (u_1,u_2) is selected to be suppressed in subcircuit II. As a result, the new element value at node (v_1,v_2), which is not suppressed, become

$$a''_{v_1,v_2} = a'_{v_1,v_2} - \frac{a'_{v_1,u_2}\Delta^{II}_{u_1,u_2}a'_{u_1,v_2}}{det(A^{II})} \tag{20}$$

We also notice that

$$det(A^{II}) = a'_{u_1,u_2}\Delta^{II}_{u_1,u_2} + R^{II}_{u_1,u_2}.$$

As a result, we have

$$\begin{aligned} a''_{v_1,v_2} &= a'_{v_1,v_2} - \frac{a'_{v_1,u_2}\Delta^{II}_{u_1,u_2}a'_{u_1,v_2}}{a'_{u_1,u_2}\Delta^{II}_{u_1,u_2} + R^{II}_{u_1,u_2}} \\ &= \frac{(a'_{u_1,u_2}a'_{v_1,v_2} - a'_{v_1,u_2}a'_{u_1,v_2})\Delta^{II}_{u_1,u_2} + a'_{v_1,v_2}R^{II}}{a'_{u_1,u_2}\Delta^{II}_{u_1,u_2} + R^{II}} \end{aligned} \tag{21}$$

Notice that $(a'_{u_1,u_2}a'_{v_1,v_2} - a'_{v_1,u_2}a'_{u_1,v_2})$ is just Eq.(13). Therefore we know that $det(A^I)$ is the common factor in the numerator and the denominator of the resulting rational function. After the common factor $det(A^I)$ is removed, a''_{v_1,v_2} will become cancellation-free.

If $det(A^I)$ is also in the denominator of $\Delta^{II}_{u_1,u_2}$, this means that more than two first-order cofactors of $det(A^I)$ are present. The cancellation will happen exactly the same way as at the middle circuit levels and $det(A^I)^k$ is still the common factor. Those cancellation will happen when all the composite terms from subcircuit I in a''_{v_1,v_2} are added together.

In our algorithm, when a matrix element consists of composite admittances from different subcircuit suppressions and those subcircuits are hierarchically related (they come from the same top level subcircuit call), we will merge them into one composite admittance and remove the common factor, $det(A^I)^k$, where $k = 1$ or 2. k will not be more than 2 because determinants or cofactors can be computed in a cancellation-free manner such that the resulting admittances are always in the correct admittance order as shown in Section 4. Such a merging or de-canceling process should have such an effect that the resulting rational composite admittance should be the same regardless of how those suppressed internal nodes are eliminated (one by one or subcircuit by subcircuit).

4. NEW S-DOMAIN HIERARCHICAL NETWORK REDUCTION ALGORITHM

In this section, we detail our network reduction algorithm based on cancellation rules discussed in early sections. In order to efficiently compute the rational function of a determinant and handle cancellations in the composite admittances, which are also rational function of s, an new DDD graph called Y-expanded DDDs (YDDD) is first introduced.

4.1. Y-expanded DDDs

Y-expanded DDDs are also DDD graphs where each DDD node represents a circuit parameter in admittance form as shown in Fig. 5.

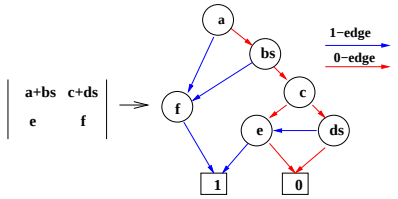

Figure 5: A determinant and its YDDD

Note that some circuit parameter admittances are function of the complex frequency variable s. This is different from the s-expanded DDD graphs where s is explicitly extracted [16]. The main propose of introduction of YDDDs is that we can easily handle aforementioned cancellation situations as this will become clear later. Similar to s-expanded DDD [16], the YDDDs can be constructed from the complex DDD in linear time in the size of the original complex DDDs and are free from the symbolic multiplication-induced cancellation from circuit devices.

4.2. Construction of Cancellation-Free Y-Expanded DDDs

Once a YDDD is constructed for a determinant, we need to compute the cancellation-free rational function from the YDDD. This can be done in a bottom up fashion. At each YDDD node P, we compute the rational function from the YDDD tree rooted at P recursively. Let $P_r = N_{P_r}/D_{P_r}$ denote the rational function of admittance that the YDDD node P *itself* represents. $P_1 = N_{P_1}/D_{P_1}$ and $P_0 = N_{P_0}/D_{P_0}$ represent the rational functions of the YDDD subtree rooted at nodes pointed by the 1-edge and 0-edge of P respectively. The rational function of the tree rooted at P will become [15]

$$\begin{aligned} F_P &= sign(P)P_r(s)P_1(s) + P_0(s) \\ &= sign(P)\frac{N_{P_r}(s)}{D_{P_r}(s)}\frac{N_{P_1}(s)}{D_{P_1}(s)} + \frac{N_{P_0}(s)}{D_{P_0}(s)}. \end{aligned} \tag{22}$$

With Eq. (22), the rational function at the root of the whole YDDD can be computed recursively. But if there are common factors among denominators of P_r, P_1 and P_0, the common factors have to be taken care of.

Specifically, we need to look at two situations where the common factors are introduced into the numerators and denominators of resulting rational functions. The first one comes from *multiplication* of $P_r(s)$ and $P_1(s)$ according to Corollary 1. The second one is due to the *addition* of two rational functions, in Eq.(22), with common factors in their denominators. Those two situations are handled separately in our method.

1. *Cancellation due to multiplication.* For cancellation from circuit devices, cancellation can be handled by dynamically removing all the canceling terms via DDD-based symbolic operations [15].
   For cancellation at middle circuit levels due to subcircuit suppressions, we observe that the new composite admittance will always have correct admittance order when all cancellations are carried out (for both cancellations between numerators and denominators, and cancellation among different product terms). Also for order-correct admittance, we notice that only $det(A^I)^k$ and $k = 1$ will be present in the denominator of the final composite admittances for each subcircuit from a circuit hierarchy, where A^I is the internal system determinant of an immediate subcircuit. This motivates a very simple cancellation strategy: *whenever we see admittance order is increased due to multiplication of two terms that both have $det(A^I)^k$ in their denominators, we divide the resulting numerator by $det(A^I)^k$.* In this way, the resulting admittance terms are always in the correct admittance order and the actual cancellation will be carried out implicitly when all the resulting terms are visited and added together. Such a *constant admittance* strategy is able to handle all the multiplication-induced cancellations without explicitly considering how the cancellation is carried out and what kind of cancellations they are.
2. *Cancellation due to addition.* For the *addition-induced* cancellation, the cancellations from circuit devices and at middle circuit levels are handled in the same way. The idea is that when adding of two rational admittances takes place, we explicitly watch for the devices which have no trivial dominators (trivial dominator is 1) such that their dominators will not appear twice in the resulting new admittance. For composite admittances, this rule requires that only one $det(A^I)$ for each subcircuit from a circuit hierarchy exists in the denominator of the resulting admittance.

Now we are ready to present the whole rational admittance computation method for a determinant represented by its YDDD tree. For each YDDD node P, we keep two things: the numerator polynomial of the computed rational admittance for the subtree rooted at P – denoted by $N(P)$ and a set of YDDD indexes whose denominators are not trivial and unique – denoted by $S_D(P)$. In this way we do not need to keep the denominator polynomial for each YDDD node explicitly and it also easy to watch the cancellation situations. $P(D)$ denotes the rational admittance for the DDD subtree rooted at D. $D.index$ is the index of the DDD node D. Following the conversion of DDD operations, we use $D.1$ and $D.0$ to represent the DDD subtrees pointed by 1-edge and 0-edge of DDD vertex D respectively. The resulting algorithm is shown Fig. 6.

The algorithm basically visits every YDDD node once in a bottom-up fashion and computes the new rational admittance for each subtree rooted at the YDDD node visited. So the time complexity of the algorithm is $O(|DDD| * k)$, where $|DDD|$ is the number of YDDD nodes and k is the highest order of all the rational admittance. It was shown that YDDD representation is more compact than *sequence of expressions* [9, 20], which essentially represent the complexity of symbolic node-by-node Gaussian elimination process. Hence time complexity of our reduction is better than $Y-\Delta$ reduction algorithm [14], which is based on Gaussian elimination.

Another advantage of the new algorithm over the existing $Y-\Delta$ reduction algorithm is that the reduction history for each node

```
COMPRAFUNC(P)
1  if ( P = zero or P = one or computed already)
2      return
3  COMPRAFUNC(P.0)
4  COMPRAFUNC(P.1)
5  N_1 = N(P.1)
6  N_0 = N(P.0)
7  if (P(P)'s denominator is not trivial and not exist in S_D(P.1))
8      S_1 = S_D(P.1) ∪ P.index
9  else
10     S_1 = S_D(P.1)
11 S_0 = S_D(P.0)
12 N_1 = sign(P) * N(P_r) * N_1
13 if (both P_r and P(P.1) contain det(A^I))
14     N_1 = N_1 / det(A^I)
15 if (both P_r and P(P.1) consist of devices with
      non-trivial denominator D_dvs)
16     N_1 = N_1 / D_dvs
17 Compute N(P) from N_1, N_0, and S_D(P) from S_1 and S_0
   such that det(A^I) and each device appear once in S_D(P).
```

Figure 6: Computation of the rational function from a YDDD.

(the neighborhood nodes reduced before the node) are not required. Our reduction process only requires the knowledge of the immediate subcircuits seen at present circuit level as the effects of all the lower level subcircuits will be merged into those immediate subcircuits and the internal system determinants $det(A^I)$ of those immediate subcircuit are the only common factors for higher level circuit reduction (except for common factors from circuit devices). Such a *reduction-history independent* feature can leads to significant memory save for reduction of large linear networks compared with $Y-\Delta$ reduction algorithm.

For de-cancellation at different circuit levels shown in Fig. 20, the *constant admittance* strategy is still followed so that the merged admittance from different subcircuits has the same admittance order by dividing the common factor $det(A^I)$.

4.3. Hurwitz Approximation To Enforce Stability

As stated in [22], Hurwitz approximation is an effective method to enforce the stability of the order-reduced transfer functions of the RLC circuit system. A polynomial is called a *Hurwitz polynomial* if all its roots have negative real parts [2]. For a given RLC system with a rational transfer function $H(s)$. $H(s)$ is stable if and only if its denominator is a Hurwitz polynomial [2].

Suppose the transfer function for a RLC system is

$$H(s) = \frac{N(s)}{D(s)} = \frac{1 + a_1 s + a_2 s^2 + \ldots + a_p s^p}{b_0 + b_1 s + b_2 s^2 + \ldots + b_q s^q}, \qquad (23)$$

Without losing generality, let q be even. We use $m(s)$ and $n(s)$ to represent the even and odd parts of $D(s)$ respectively:

$$m(s) = b_0 + b_2 s^2 + \ldots + b_q s^q, \qquad (24)$$

$$n(s) = b_1 + b_3 s^3 + \ldots + b_{q-1} s^{q-1}. \qquad (25)$$

Then the fraction $n(s)/m(s)$ can be written as the following form:

$$\frac{n(s)}{m(s)} = \frac{1}{c_0 s^{-1} + \cfrac{1}{c_1 s^{-1} + \cfrac{1}{\ldots + \cfrac{1}{c_{q-1} s^{-1}}}}}, \qquad (26)$$

where $D(s)$ is said to be a Hurwitz polynomial if the coefficients c_0, c_1, ...,c_{q-1} are all positive. To find a kth-order Hurwitz approximation $H_k(s)$ for a given transfer function $H(s)$, we can keep all the positive coefficients c_0 up to c_k and ignore all the other coefficients, and then re-calculate the odd and even parts of the denominator $D_k(s)$ of $H_k(s)$ according to equation (26). Once we obtain the stable new denominator $D_k(s)$, the new numerator, $N_k(s)$, in the final approximated rational function $H_k(s) = \frac{N_k(s)}{D_k(s)}$ can be obtained with explicit moment matching by which the first k moments are equivalent in $H_k(s)$ and $H(s)$, or $H(s) - H_k(s) = O(s^k)$. The new transfer function $H_k(s)$ is the kth-order Hurwitz approximation of the original system and is guaranteed to be stable. Note that the explicit moment matching here is numerical stable for very high orders as no ill-conditioned matrix is solved as is the case for AWE [12].

4.4. Overview of The Reduction Algorithm

After describing several critical aspects of the reduction algorithm, we are ready to give an overview of the whole reduction algorithm in this subsection. We assume that the circuit hierarchy has been defined already.

The General Hierarchical Network Reduction Algorithm

1. Build the complex DDDs for each subcircuit system matrix and all the required cofactors in a bottom up way.
2. If the present circuit has subcircuits, perform the reduction on each subcircuit first.
3. In the present circuit, identify all the admittances, which are either from circuit parameters or are composite admittance due to subcircuit suppression.
4. Merge all the composite admittances from subcircuits at different circuit hierarchical levels and remove common factors by dividing the numerators by $det(A^I)$, which is the internal system determinant of an immediate subcircuit at lower circuit level. The merging process is done from the lowest level to the highest level.
5. Derive the YDDDs from complex DDDs for the internal system determinant and all the cofactors for upper level circuits or for required transfer function using COMPRAFUNC()
6. Perform Hurwitz approximation on the reduced transfer functions.

5. EXPERIMENTAL RESULTS

The proposed algorithm is implemented in C++. We perform the reduction on a number of linear analog circuits. Here we present the exact transfer function for a second order linear filter shown in Fig. 7 obtained from our reduction program. In this circuit, we have $R_1 = 1k\Omega$, $R_2 = 2k\Omega$, $R_3 = 5k\Omega$, $C_1 = 10nF$, $C_2 = 20nF$, A_0 is an Opamp circuit, a simple implementation of it is shown in Fig. 8. The transfer function obtained by our program is shown in the following:

$$H(s) = \frac{V_{out}(s)}{V_{in}(s)} = \frac{\frac{5.33\times10^{-7}+2.68\times10^{-15}s+3.72\times10^{-19}s^2}{1.02\times10^{-3}+2.66\times10^{-6}s}}{\frac{2.66\times10^{-7}+9.07\times10^{-11}s+5.46\times10^{-16}s^2+7.47\times10^{-23}s^3}{1.02\times10^{-3}+2.66\times10^{-6}s}}$$

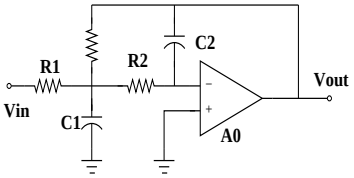

Figure 7: A second-order active filter

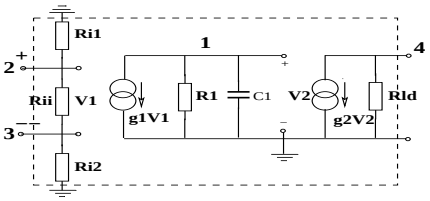

Figure 8: A linear model of an Opamp circuit.

Note that both the numerator and the denominator become rational functions of s due to suppression of subcircuit A_0. But they share a common denominator which is actually the internal system determinant – $det(A^{A_0})$ of the subcircuit A_0. After the common denominator is removed, the transfer function is exactly the same one as we obtain from the flatten filter circuit. We also notice that the highest order of the transfer function of the second order filter becomes 3. But if Opamp A_0 is a ideal voltage controlled current source with infinite gain, the highest order will be 2.

We also notice that since the admittance orders (the number of admittances of circuit devices in the denominator) of new admittances are preserved, the numerical magnitudes of the coefficients at varying orders of s in both numerators and denominators of the new admittance will grow or decrease very fast with number of nodes reduced. As a result, proper scaling of input data is required for reduction of very large linear networks.

Second, we test the proposed algorithm on a number of RLC circuits, some of the circuits are extracted clock wires and bus structure coupled wires. The experimental results are obtained on a Linux workstation with dual 1GHz P-III CPUs and 2G memory. In our examples, 15th order is the highest order kept for the generated admittances.

The statistics are summarized in the table 1. The first column gives the names of the circuits, *ckt1* to *ckt4* are general RC circuits with many coupling capacitors; *ckt_tree5* and *ckt_tree6* are RCL tree circuits. Column *#nodes* gives the number of nodes in each circuit. To obtain the responses from SPICE, the largest number of nodes in a circuit is about 80k. Column *#level* shows the highest hierarchical level in each circuits. The hierarchical structures are obtained via an automatic partitioning algorithm [19]. Column *#C* and *#L* are the number of capacitors and inductors in each circuit. Column *new method (CPU)* is the CPU time for the new algorithm to perform the reduction and to perform the s-domain analysis on the reduced network on 100 sampling points. The last column is the CPU time used by SPICE for the same s-domain analysis task.

From Table 1, we can observe that the new reduction algorithm is one or two order of magnitudes faster than Spice3f4. It can be logically expected that the speedup will be even more significant for larger RLC circuits.

Figure 9 gives the responses for different orders of Hurwitz approximations for a small coupled RLC circuit where we can obtain the exact transfer functions. As the approximation order increases,

Table 1: Statistics for some RLC circuits

Ckt	#nodes	#level	#C	#L	new method (CPU)	SPICE (CPU)
ckt1	90	3	120	0	0.26	0.03
ckt2	762	6	1016	0	0.76	0.28
ckt3	12282	10	16376	0	7.3	47.25
ckt4	49146	12	65528	0	27.3	816.15
ckt_tree5	20470	10	10235	10235	2.74	388.45
ckt_tree6	81910	12	40955	40955	10.32	717.720

the approximation gets better.

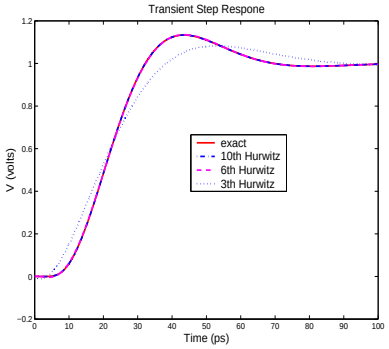

Figure 9: Transient step responses given by the new reduction algorithm for different order Hurwitz approximation for a couple RLC circuit.

6. CONCLUSION

In this paper, we have proposed a general hierarchical linear network reduction technique in s-domain. The reduction is done by subcircuit suppression in a hierarchical way and by rational function approximation, which generates order-reduced cancellation-free admittances in the reduced network matrices. The new method works on circuit matrices directly and can be applied to any linear circuits. On the theoretical side, we studied how common factors are generated in the general subcircuit reduction process and presented some theoretical results. On the practical side, we proposed a novel de-cancellation strategy based on determinant decision diagrams to derive the order-reduced rational functions for determinants and cofactors generated from subcircuit reductions in a cancellation-free manner. The stability is enforced by Hurwitz approximation on the order-reduced transfer functions. Experimental results validate the proposed method on some linear analog circuits and large RLC circuits. For the given RLC linear circuits, the new method has shown to offers two orders of magnitude speedup over Spice3f4 in s-domain analysis.

ACKNOWLEDGEMENT

The author would like to thank Dr. Zhanhai Qin and Prof. Chung-Kuan Cheng for discussions on the $Y-\Delta$ transformation algorithm, Junjie Yang for implementation of Hurwitz approximation and anonymous reviewers for their constructive comments. This work is partially funded by University of California Academic Senate Research Fund.

7. REFERENCES

[1] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*, Reading, Massachusetts:Addison-Wesley Publishing Company, 1990.

[2] C.T. Chen, "Linear System Theory and Design", *Oxford Univ. Press*, 3rd Ed., 1999.

[3] P.J.H. Elias, N.P. van der Meijs, "Including higher-order moments of RC interconnections in layout-to-circuit extraction" in *Proc. European Design and Test Conference (ED&TC)*, pp.362-6. 1996.

[4] P. Feldmann and R. W. Freund, "Efficient linear circuit analysis by Pade approximation via the Lanczos process", *IEEE Trans. Computer-Aided Design*, vol. 14, no. 5, pp. 639-649, Feb. 1995.

[5] —, "Reduced-order modeling of large linear subcircuits via block Lanczos algorithm", in *Proc. IEEE/ACM Design Automation Conf.(DAC)*, pp. 376–380, June. 1995.

[6] —, "Reduced-order modeling of large linear subcircuits by means of the SyPVL algorithm", in *Proc. IEEE Int. Conf. Computer-Aided Design (ICCAD)*, pp. 280–287, Nov. 1996.

[7] J. Cong, L He, C.-K. Koh and P. Padden "Performance Optimization of VLSI Interconnect Layout", *Integration, the VLSI Journal*, vol. 21, nos. 1 & 2, Nov. pp. 1–94, 1996.

[8] C. K.Cheng, J. Lillis, S. Lin and N. H. Chang, *Interconnect Analysis and Synthesis*, New York:John Wiley & Sons, Inc., 2000.

[9] M. M. Hassoun and P. M. Lin, "A hierarchical network approach to symbolic analysis of large scale networks", *IEEE Trans. Circuits and Systems*, vol. 42, no. 4, pp. 201–211, April 1995.

[10] Y. Ismail and E. G. Friedman "DTT: direct truncation of the transfer function – an alternative to moment matching for tree structured interconnect", *IEEE Trans. Computer-Aided Design*, vol. 21, no. 2, pp. 131-144, Feb. 2002.

[11] A. Odabasioglu, M. Celik and L. T. Pilagge, "Prima: passive reduced-order interconnect macromodeling algorithm," *IEEE Trans. on Computer-Aided Design*, vol. 17, no. 8, pp. 645-654, Aug. 1998.

[12] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 352-366, Apr. 1990.

[13] L. T. Pillage, R. A. Rohrer and C. Visweswariah *Electronic Circuit and System Simulation Methods*, New York, McGraw-Hill, 1994.

[14] Z. Qin and C.-K. Cheng, "RCLK-VJ network reduction with Hurwitz polynomial approximation," *Proc. Asia South Pacific Design Automation Conference (ASP-DAC)*, pp.283–291, Jan. 2003.

[15] C.-J. Shi and X.-D. Tan, "Canonical symbolic analysis of large analog circuits with determinant decision diagrams", *IEEE Trans. Computer-Aided Design*, vol. 19, no. 1, pp. 1–18, Jan. 2000.

[16] C.-J. Shi and X.-D. Tan, "Compact representation and efficient generation of s-expanded symbolic network functions for computer-aided analog circuit design", *IEEE Trans. Computer-Aided Design*, vol. 20, No. 7, pp. 813-827, July 2001.

[17] M. Silveira, M. Kamon, and J. White, "Efficient reduced-order modeling of frequency-dependent coupling inductances associated with 3-D interconnect structures", in *Proc. IEEE/ACM Design Automation Conf.(DAC)*, pp. 376–380, June 1995.

[18] M. Silveira, M. Kamon, I. Elfadel and J. White, "A coordinate-transformed Arnoldi algorithm for generating guaranteed stable reduced-order models of RLC circuits", in *Proc. IEEE Int. Conf. Computer-Aided Design (ICCAD)*, pp. 288–294, Nov. 1996.

[19] X.-D Tan and C.-J. Shi, "Balanced multi-level multi-way partitioning of large analog circuits for hierarchical symbolic analysis," *Proc. Asia and South Pacific Design Automation Conference (ASP-DAC'99)*, pp. 1-4, Jan., 1999.

[20] X.-D. Tan and C.-J. Shi, "Hierarchical symbolic analysis of large analog circuits via determinant decision diagrams", *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 4, pp. 401-412, April 2000.

[21] S. X.-D. Tan and J. Yang, "Hurwitz stable model reduction for non-tree structured RLCK circuits," in *Proc. 16th IEEE International System-on-Chip Conference*, Sept, 2003. (to appear).

[22] X. Yang, C.-K. Cheng, W.H. Ku, R. J. Carragher "Hurwitz stable reduced order modeling for RLC interconnect trees", in *Proc. IEEE Int. Conf. Computer-Aided Design (ICCAD)*, pp. 222–228, Nov. 2000.