

RCLK-VJ Network Reduction With Hurwitz Polynomial Approximation

Zhanhai Qin
email: zqin@cs.ucsd.edu

Chung-Kuan Cheng
email: kuan@cs.ucsd.edu

Department of Computer Science & Engineering
University of California, San Diego
9500 Gilman Drive, La Jolla, CA, 92093, U.S.

Abstract— We propose a new linear network reduction algorithm based on a generalized $Y-\Delta$ transformation technique in s -domain. Resultant admittance is kept as a rational function of s with a dramatically reduced order. Yet it preserves low-order terms of exact admittance evaluated with traditional symbolic analysis. Stability of transfer functions derived from reduced-order admittance is guaranteed via a Hurwitz polynomial approximation. Such low-order transfer functions are used in pole analysis and time domain waveform evaluation in response to any input signal.

I. INTRODUCTION

Linear networks existing in modern VLSI chips include power/ground meshes, clock distribution networks, and global interconnects. Because these networks tend to contain millions of lumped linear RCL elements, simulating them in general tools such as SPICE is not practical. Instead, two strategies are widely adopted: (1) to increase the efficiency of solving linear simultaneous equations; and (2) to reduce the size of original networks using model order reduction techniques.

In the first category, preconditioned Krylov-subspace iterative methods[1] with Nodal Analysis (NA) are shown more efficient than LU factorization with Modified Nodal Analysis (MNA) in SPICE. [3] shows that SuperLU factorization[2] with NA provides comparable performance to iterative methods while the robustness of direct methods is kept. [4, 5] explore the regular grid structure of power/ground networks and use multigrid technique to solve a coarse grid and map the solution back to the original fine grid.

In the second category, the moment-matching technique has been used to approximate waveforms of a linear interconnect network using its lower order moments[6, 7, 8]. Since the advent of Asymptotic Waveform Evaluation(AWE) technique, many interconnect delay evaluation models [9, 10, 11] have been proposed. It is well known that Padé approximation used in AWE may generate positive poles for an originally passive system.

Because of the drawback of the moment-matching method, [12] proposes a method to realize reduced RC sub-networks as macromodels. Sub-networks are reduced by means of reserving lower orders of the port admittance matrix. The method guarantees the realizability of the macromodels for RC cir-

cuits. Matrix Padé Via Lanczos[13], block Arnoldi[14] and PRIMA[15] are admittance-matrix-based model order reduction methods, so that they perform model order reduction on each entry in admittance matrix simultaneously. The PACT algorithm[16] first introduces congruence transformations for order reduction of RC circuits. The same authors propose split congruence transformations[17] for passive reductions of RCL circuits.

In another aspect, topological analysis [18] is an approach to calculating driving-point admittances using Cramer's rule in s -domain. The determinant of an admittance matrix of a passive network without mutual inductances is equal to the sum of all the tree admittance products of the network. The advantage of topological analysis formula over conventional methods evaluating determinants is that it avoids the usual cancellations inherent in the expansion of determinants in the latter. But enumerating all the trees in a large network is impractical.

Recently [19] proposes a direct transfer-function truncation (DTT) method to approximating transfer functions in tree-structured RCL networks in s -domain. The transfer functions are kept in rational expressions in s , and an approximation is acquired by directly truncating high-order terms. Such an approximation also matches low-order time moments implicitly, but truncated characteristic denominator may not be stable any more. The method is able to obtain very high-order transfer functions when AWE fails because of numerical problems.

We have proposed a new RCL network reduction method. The principal idea is that we consider a linear network as a graph and perform $Y-\Delta$ transformation[20] on each node of no interest in the graph, until all such nodes are eliminated. After each transformation, any admittance of order higher than a threshold value will be truncated. The truncation would result in a low-order admittance, an approximation to the exact one. Different from topological analysis and other traditional symbolic analysis, our approach keeps only coefficients of low-order terms after the transformation. These coefficients, however, are exactly the same as those in exact admittance. $Y-\Delta$ transformation is further generalized to handle current/voltage sources and K elements[21]. In this paper, nodes eligible to be eliminated are called *internal* nodes. and others are called *external* nodes.

Generally speaking, the input admittance of an single-terminated N -th order RCL linear time invariant network in

s -domain is a N -th order rational function $Y(s)$. Y- Δ transformation reduces the network in term of the number of nodes, but a straightforward implementation of the approach, however, leads to a rational function $Y'(s)$ whose order would be far beyond N (exponential of N). It is worth noting that $Y(s) = Y'(s)$ indeed. By exploiting the structure of Y- Δ transformation process, we have found out that many common factors are introduced into the numerator and denominator of $Y'(s)$, which actually should have been canceled out. This finding, together with some other practical numerical considerations, allow us to control round-off errors in higher-order polynomial computation. It is helpful in pole/zero approximation because otherwise these common factors will be mixed with system poles/zeros.

Our main contributions are:

1. admittance is always kept in its rational form and all the coefficients are the same as they were computed using exact symbolic approaches without discarding any high-order terms;
2. First $K + 1$ time moments are matched implicitly, including constant term m_0 ;
3. Common factor effects are found in Y- Δ transformation for linear networks. The finding improves numerical stability of the approach, and leads to a more accurate pole/zero approximation.
4. A Hurwitz polynomial approximation is employed on truncated transfer functions of Y- Δ transformation, so that a stable reduction is guaranteed.

The remaining of the paper is organized as follows. In the next section, we briefly review the background knowledge of Y- Δ transformation. A generalized formula is given in Section III. Common factor effects are shown and detailed in Section IV. In Section V, a node ordering algorithm, revised Modified Multiple minimum Degree (MMD)[24] is described. And Section VI is dedicated to the explanation of the overall reduction flow. Section VIII shows our experimental results, and Section IX concludes the proposed algorithm.

II. BACKGROUND

First we illustrate a numerical example on Y- Δ transformation. In the circuit shown in Fig. 1(a), n_0 is adjacent to n_1 , n_2 , and n_3 . Kirchhoff's Current Law (KCL) equations for node n_0 , n_1 , n_2 , and n_3 can be established as follows:

$$(6 + 7s)V_0 - 7sV_1 - V_2 - 5V_3 = I_{g_0} \quad (1)$$

$$-7sV_0 + 7sV_1 = 0 \quad (2)$$

$$-V_0 + V_2 = 0 \quad (3)$$

$$-5V_0 + 5V_3 = 0. \quad (4)$$

From (1), we can denote V_0 in terms of V_1 , V_2 , and V_3 as:

$$V_0 = \frac{7sV_1 + V_2 + 5V_3 + I_{g_0}}{6 + 7s}. \quad (5)$$

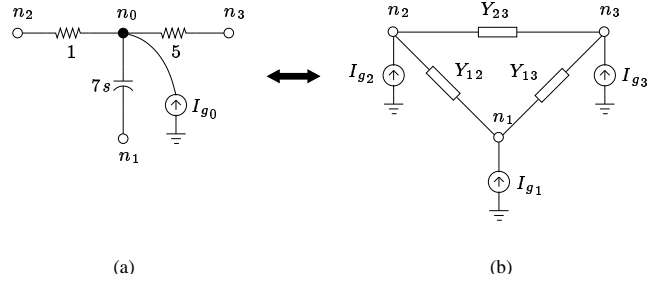


Fig. 1. A numerical example for Y- Δ transformation: (a)circuit schematic before the transformation; (b)circuit schematic after the transformation.

Inserting (5) into (2)–(4) yields

$$\frac{42s}{6 + 7s}V_1 - \frac{7s}{6 + 7s}V_2 - \frac{35s}{6 + 7s}V_3 = \frac{7s}{6 + 7s}I_{g_0} \quad (6)$$

$$-\frac{7s}{6 + 7s}V_1 + \frac{5 + 7s}{6 + 7s}V_2 - \frac{5}{6 + 7s}V_3 = \frac{1}{6 + 7s}I_{g_0} \quad (7)$$

$$-\frac{35s}{6 + 7s}V_1 - \frac{5}{6 + 7s}V_2 + \frac{5 + 35s}{6 + 7s}V_3 = \frac{5}{6 + 7s}I_{g_0}. \quad (8)$$

Considering (6)–(8) as KCL equations for n_1 , n_2 , and n_3 in Fig. 1(b), respectively, we can find out that

$$\begin{cases} Y_{12} = \frac{2s}{6+7s} \\ Y_{13} = \frac{3s}{6+7s} \\ Y_{23} = \frac{6}{6+7s} \end{cases} \quad \text{and} \quad \begin{cases} I_{g_1} = \frac{7s}{6+7s}I_{g_0} \\ I_{g_2} = \frac{1}{6+7s}I_{g_0} \\ I_{g_3} = \frac{5}{6+7s}I_{g_0}. \end{cases} \quad (9)$$

From the example, we notice that what we performed is equivalent to one Gauss elimination to the system equations. When actual Y- Δ transformations are carried out, however, we do not formulate a circuit into simultaneous system equations. Instead we consider it as a graph and operate on the graph directly.

III. GENERALIZED Y- Δ TRANSFORMATION

The traditional Y- Δ transformation is generalized in this paper to handle RCLK passive elements, independent voltage sources (V), and independent current sources (J). The traditional Y- Δ transformation allows RCL as branch elements, because they have well-known admittance forms in s -domain. But Y- Δ transformations involving mutual K elements[21] and voltage/current sources are not straightforward.

K-based inductance extraction method proposes the new circuit element — K to capture the inductance effects of interconnects in integrated circuits. (14) of [22] gives the branch equation for element K:

$$Kv = \frac{di}{dt}, \quad (10)$$

which in s -domain can be written as:

$$\frac{K}{s}V = I. \quad (11)$$

Although V and I refer to different branches for capturing mutual inductance effects, a simple conversion will integrate K elements into our transformation formula seamlessly. For the sake of simplicity, in our presentation we assume that all storage elements here have no initial conditions. Initial conditions are simply modeled in s -domain as constant current or voltage sources.

A. Branch with K element

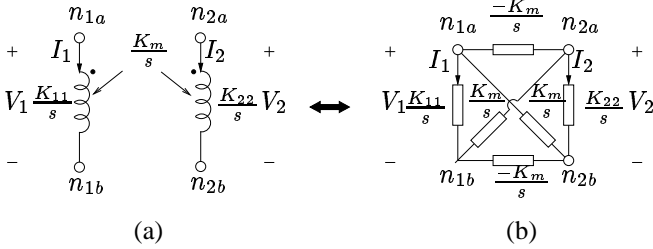


Fig. 2. Conversion on mutual K in s -domain: (a) mutual K element given; (b) converted self K elements.

In Fig. 2(a) the circuit branch equations can be written as

$$\begin{cases} \frac{K_{11}}{s} V_1 + \frac{K_m}{s} V_2 = I_1 \\ \frac{K_m}{s} V_1 + \frac{K_{22}}{s} V_2 = I_2. \end{cases} \quad (12)$$

One can check that the branch equations for the two branches in Fig. 2(b) are exactly the same as (12), so that the circuit in (b) is equivalent to the one in (a). Although some values in (b) are negative, the equivalent circuit is still passive because K -based method guarantees the extracted K matrix to be positive definite.

B. Generalization

We state a generalized Y - Δ transformation formula including linear current/voltage sources, resistors, capacitors, self partial inductors and K elements.

Theorem 1 *With no loss of generality, let n_0 be the node that to be eliminated, let n_1, n_2, \dots, n_k be the adjacent nodes to n_0 . Y_{ij} denotes the admittance between node n_i and n_j . Thus $Y_{01}, Y_{02}, \dots, Y_{0k}$ are the admittance between n_0 and n_1, n_2, \dots, n_k , respectively. Particularly, a current source is considered to be open-circuited and a voltage source short-circuited in terms of admittance. In s -domain, admittance is a function of s .*

After n_0 is eliminated, n_1, n_2, \dots, n_k become pairwise adjacent and form a clique. A set of admittance

$$\{Y_{ij} \mid i, j \in [1, k], i < j\}$$

are generated, and

$$Y_{ij}(s) = Y_{ij}(s) + \frac{Y_{0i}(s) \times Y_{0j}(s)}{Y_{01}(s) + Y_{02}(s) + \dots + Y_{0k}(s)}. \quad (13)$$

Suppose I_{01} was a current source between n_0 and n_1 before the elimination of n_0 . A set of current sources

$$\{I_{1j} \mid j \in [2, k]\}$$

are to be generated after the elimination, and

$$I_{1j}(s) = \frac{Y_{0j}(s)}{Y_{01}(s) + Y_{02}(s) + \dots + Y_{0k}(s)} \times I_{01}(s). \quad (14)$$

Voltage sources can be transformed to current sources before any elimination begins. Suppose V_{01} was a voltage source between n_0 and n_1 before the transformation. The transformation of the voltage source results in a set of current sources,

$$\{I_{1j} \mid I_{1j}(s) = Y_{0j}(s)V_{01}(s), j \in [2, k]\}.$$

A useful observation from Th. 1 is that different from AWE method, in Y - Δ transformation, coefficients of admittance are derived directly from admittance in original circuits and are kept in its original rational form. By matching the lower-order coefficients, stable reduced-order models can be obtained in Section VII.

Corollary 1 *If all RCL elements in a given linear RCL system are of positive values, no matter how many nodes are eliminated via Y - Δ transformation, the transformed admittance between any two nodes n_i and n_j can be written as*

$$Y_{ij}(s) = \frac{\sum_{p=0}^m a_p s^p}{\sum_{q=0}^n b_q s^q}, \quad a_p, b_q \geq 0. \quad (15)$$

The above corollary holds immediately after Th. 1.

IV. COMMON FACTOR EFFECT

There are two kinds of common factors that have been introduced to the generalized Y - Δ transformation. The first is common factors among denominators of admittance, and the other is common factors in both numerators and denominators. We are able to prove that they are from the same source — nodes eliminated. Common-factor effects are harmful to our reduction algorithm because: (1) it causes the magnitude of coefficients of the numerator and denominator unnecessarily wildly grow; (2) common factors in numerators/denominators create fake zeros/poles that hamper a pole/zero analysis.

Y - Δ transformation is a continuous process, and the network topology, as well as branch admittance, will change dynamically along with the process. We denote the network after the t -node is eliminated as graph $G^{(t)}$, and the original network as $G^{(0)}$. The admittance of branch $e_{i,j}$ in the original graph as $Y_{i,j}^{(0)}$, and $Y_{i,j}^{(1)}$ for branch $e_{i,j}$ in the subsequent graph when

the first node is eliminated and so on. In this way, we can rewrite (13) into

$$Y_{i,j}^{(t)}(s) = Y_{i,j}^{(t-1)}(s) + \frac{Y_{t-1,i}^{(t-1)}(s) \times Y_{t-1,j}^{(t-1)}(s)}{Y_{t-1,1}^{(t-1)}(s) + Y_{t-1,2}^{(t-1)}(s) + \dots + Y_{t-1,k}^{(t-1)}(s)} \quad (16)$$

for the t -th transformation.

Definition 1 Given a node n_k with m neighbors in a graph from an on-going Y - Δ transformation process, we denote admittance between n_k and its neighbors using $\frac{a_1}{b_1}, \frac{a_2}{b_2}, \dots, \frac{a_m}{b_m}$. We define

$$\omega_k = \frac{\sum_{i=1}^m \left(a_i \prod_{j=1, j \neq i}^m b_j \right)}{W_k}, \quad (17)$$

where

$$W_k = \prod_{i=1}^m \omega_i^{p_i-1} \quad (18)$$

and p_i is the number of denominators in $\{b_1, \dots, b_m\}$ carrying factor ω_i .

ω is the atomic common factor. The first kind of common factors is easy to identify. Based on (13), ω defines a factor that denominators of new admittance will be sharing after n_k is eliminated. This factor exists in these denominators even though admittance merging could take place from time to time, when nodes other than its ex-neighbors are eliminated. This is because admittance merging is an addition operation of two rational functions and the denominator of the resultant admittance is a multiplication of the operands' two denominators. Thus when one of n_k 's ex-neighbors is to be eliminated, this factor should be recognized as a common factor among denominators of admittance of present branches incident to the ex-neighbor node. The observation is true for any nodes with more than two neighbors, and we generalize it as the following theorem. With no loss of generality, we assume that n_0 is the node to be eliminated.

Theorem 2 n_0 is the node to be eliminated with $k(k > 2)$ neighbors in a graph from an on-going Y - Δ transformation process. Suppose L_0 is a set of its neighbors. $n_i \in L_0$ is assumed to be the first node among those in L_0 in the elimination sequence behind n_0 . There may be other nodes between n_0 and n_i in the sequence. When n_i is to be eliminated, denominators of branch admittance incident to n_i share a factor ω_0 , which is associated with n_0 as defined in Def. 1.

The other kind of common factors are shared between the numerator and denominator of admittance. Their existence in denominators is apparent. To account for their existence in numerators, we have to prove for their first appearance in numerators(Th. 3), followed by a proof (Th. 4) that cancellation of any common factors between numerators and denominators does not affect others' appearance in numerators of upcoming admittance. A rigorous proof can be found in [26].

Theorem 3 n_0 is the node to be eliminated with $k(k > 2)$ neighbors in a graph $G_i(V_i, E_i)$ from an on-going Y - Δ transformation process. Suppose L_0 is a set of its neighbors. And $n_i \in L_0$ is the first node among those in L_0 in the elimination sequence behind n_0 . There may be other nodes between n_0 and n_i in the sequence. After n_0 and n_i are eliminated, numerators of branch admittance incident to any two nodes in $L_i - \{n_i\}$ has a factor ω_0 , which is associated with n_0 as defined in Def. 1.

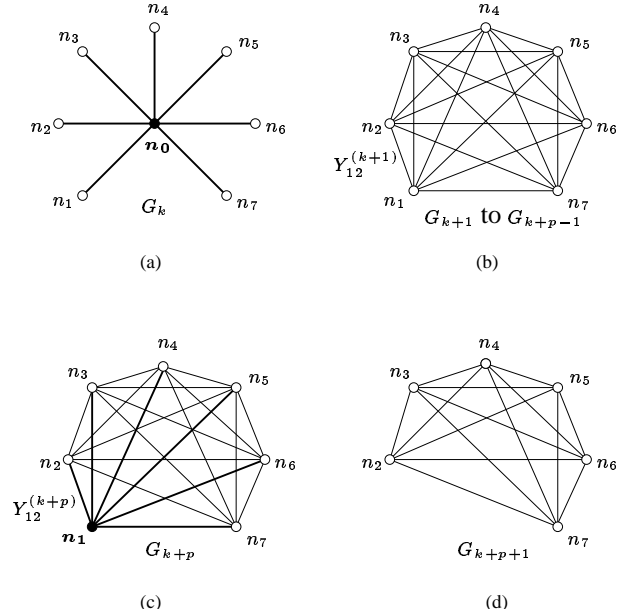


Fig. 3. Illustration for Theorem 4: (a) k -th transformation; (b) $(k+1)$ -th transformation; (c) $(k+p)$ -th transformation; (d) $(k+p+1)$ -th transformation.

Theorem 4 n_0 is the node to be eliminated in graph G_k at the k -th step of an on-going Y - Δ transformation process(Fig. 3(a)). Let L_0 represent the neighbors of n_0 in G_k . And $n_1 \in L_0$ is the first node among those in L_0 in the elimination sequence behind n_0 . There may be other nodes between n_0 and n_i in the sequence, as shown in Fig. 3(b).

Suppose n_1 is to be eliminated in G_{k+p} (Fig. 3(c)). numerators of new admittance between any two nodes of n_1 's neighbors in Fig. 3(c) will have a factor ω_0 which is associated with n_0 as defined in Def. 1, no matter how many other common factors are found and canceled out during the $(k+1)$ -th step through the $(k+p)$ -th step (suppressed as one step in Fig. 3(b)).

Similar to mathematical induction, Th. 3 assures the foundation of our reduction algorithm, and Th. 4 makes our reduction process work recursively. The two theorems together support the overall reduction algorithm in Section VI.

V. NODE ORDERING

The order of picking nodes to eliminate is important, because eliminating nodes in a network via Y- Δ Transformation is equivalent to LU factorizing the corresponding MNA formulated linear equations. Non-zero fill-ins in LU factorization corresponds to new branches in the reduced network. Therefore different elimination orders will result in a different number of new branches. And the complexity of Y- Δ transformation on a node n_i of degree d is $O(d^2)$. We employ an ordering scheme in sparse matrix computation: MMD algorithm.

The most widely used general-purpose ordering scheme is the minimum-degree algorithm [23]. Given a simple graph, the node with the minimum degree is eliminated from the graph and degrees of affected nodes are updated, and the node with the new minimum degree in the new graph is taken next... It is used as a practical approximate solution to the NP-complete fill minimization problem [25].

The concept of indistinguishable nodes [23] was developed to eliminate a subset of nodes all at the same time instead of just one node at a time. In the elimination process, nodes n_i and n_j that satisfy

$$\text{Adj}(n_i) \cup \{n_i\} = \text{Adj}(n_j) \cup \{n_j\}$$

in a graph are said to become indistinguishable. $\text{Adj}(n_i)$ means the set of n_i 's neighbors. These nodes can be numbered consecutively in the minimum-degree ordering.

Algorithm 1. Node Ordering

1. (Initialization) Initialize the set of uneliminated nodes $X = \{n_i | n_i \text{ is an internal node}\}$. Compute the degree of all the nodes in X .
2. (Selection) Pick a node n with the minimum degree.
3. (Mass Elimination) Number the node n and those indistinguishable from n .
4. (Degree update) Determine the representation of the new graph. Update the degrees of n 's neighbors.
5. (Loop or Stop) Repeat steps 2–4 until all nodes are eliminated.

Theorem 5 Let S_1 and S_2 denote any different node elimination sequences for a given network. Suppose n_i and n_j are two external nodes of the circuit. Let $Y_{i,j}^{(n)}(s)$ and $Y_{i,j}^{(m)}(s)$ are the input admittance of port n_i – n_j after Y- Δ transformation by following elimination sequence S_1 and S_2 , respectively. The following equation holds:

$$Y_{i,j}(s) = Y_{i,j}^{(n)}(s).$$

The theorem tells us that even though different node elimination sequences could have dramatically different impact on the performance of reduction, the transformed admittance is independent from them.

Generally speaking, input admittance of any two nodes in a non-degenerated network (no loop capacitors or cut-set inductors) with n lumped capacitors and/or inductors and any amount of resistors is a n -th rational function of s . Even though n could be huge, we only need to keep coefficients of $Y_{i,j}^{(k)}$'s lower order terms, i.e., a_i and b_j of Y_{ij} in (15). Th. 6 ensures us that for any transformed admittance $Y_{i,j}^{(k)}$ at any step of a Y- Δ transformation process, keeping its lower t order coefficients in its numerator and denominator throughout the whole reduction process will make the final transformed admittance be a truncation of the exact admittance.

Theorem 6 With no loss of generality, let us refer to (13). Suppose we have two Y- Δ reduction procedures A and B following the same elimination sequence. At the k -th step, a newly transformed admittance in A is termed as $Y_{i,j}^{(k)}$ and can be computed as

$$Y_{i,j}^{(k)}(s) = \frac{Y_{i0}^{(k-1)}(s) \times Y_{j0}^{(k-1)}(s)}{\sum_{i=1}^t Y_{i0}^{(k-1)}(s)}.$$

At the k -th step in B, a newly transformed admittance is termed as $\tilde{Y}_{i,j}^{(k)}$ and can be computed as

$$\tilde{Y}_{i,j}^{(k)}(s) \approx Y'^{(k)}_{ij}(s) = \frac{\tilde{Y}_{i0}^{(k-1)}(s) \times \tilde{Y}_{j0}^{(k-1)}(s)}{\sum_{i=1}^t \tilde{Y}_{i0}^{(k-1)}(s)}.$$

Here $Y'^{(k)}_{ij}$ is in the form

$$Y'^{(k)}_{ij}(s) = \frac{a_0 + a_1 s + \dots + a_m s^m}{b_0 + b_1 s + \dots + b_n s^n},$$

and $\tilde{Y}_{ij}^{(k)}$ is the k -th order approximate of $Y'^{(k)}_{ij}$

$$\tilde{Y}_{ij}^{(k)}(s) = \frac{a_0 + a_1 s + \dots + a_t s^t}{b_0 + b_1 s + \dots + b_t s^t}, \quad 0 \leq t \leq \min(m, n).$$

If $\tilde{Y}_{i0}^{(k-1)}$, $\tilde{Y}_{j0}^{(k-1)}$, $\tilde{Y}_{10}^{(k-1)}$, $\tilde{Y}_{20}^{(k-1)}$, \dots , $\tilde{Y}_{t0}^{(k-1)}$ are the t -th order approximate of $Y_{i0}^{(k-1)}$, $Y_{j0}^{(k-1)}$, $Y_{10}^{(k-1)}$, $Y_{20}^{(k-1)}$, \dots , $Y_{t0}^{(k-1)}$, respectively, then $\tilde{Y}_{ij}^{(k)}$ is also the t -th order approximate of $Y_{i,j}^{(k)}$.

The theorem can be proven using mathematical induction. Note that the result is based on Y- Δ transformation with consideration of common factor cancellation.

VI. REDUCTION FLOW

Alg. 2 describes a linear circuit reduction algorithm using generalized Y- Δ transformation. There are three major steps in the algorithm: the first step is to pre-process the given circuit to make sure there is no voltage sources or floating current sources; the second step is to call node ordering algorithm to generate an order of picking nodes to eliminate; and finally the

third step is to pick nodes one by one upon the order and perform Y- Δ transformation.

Algorithm 2. Reduction Flow

1. (Source Transformation and Decoupling) Transform voltage sources to current sources, and decouple any floating current source.
2. (Node Ordering) Call Alg. 1 to generate a node elimination sequence S .
3. (Node Elimination) For $n_i = \text{First}(S)$
 - 3.1. Compute new admittance using formula in (13), with consideration of common factors in denominators. Computation of terms higher than the threshold value will be skipped.
 - 3.2. Cancel common factors, if any, shared between the numerator and denominator in new admittance.
 - 3.3. If n_i has a current source, compute new current sources for its neighbors using (14).
 - 3.4. RemoveFirst(S).

A current source is said to be *floating* when it flows from one non-datum node to another non-datum node. Decoupling it is to remove the current source, and insert two concatenated ones of the same amount of current between the two end-nodes. They are concatenated at the ground node. Through the equivalent source transformation, decoupled sources become associated with nodes instead of branches and make our algorithm simpler.

The algorithm takes $O(N \cdot D^2 \cdot r^2)$, where N is the number of nodes in an elimination sequence, D is the maximum degree of nodes in G_0, \dots, G_{N-1} , and r is the threshold value indicating the number of orders being preserved for each admittance. Because each rational addition operation takes r^2 scalar multiplications and additions, so r^2 is in the formula. The estimation is very conservative, because only a few percent of nodes have degrees near the upper bound D , while most nodes are much less than it.

Different from LU decomposition in SPICE, our algorithm allows dynamical memory de-allocation, as branches of nodes eliminated are no longer needed and can be freed. As a result, the memory requirement grows up in the middle of the reduction process and goes down when it ends. The peak memory consumption is proportional to the maximum number of branches in graphs G_0, G_1 , etc. in a reduction process.

VII. HURWITZ POLYNOMIAL APPROXIMATION

A Hurwitz polynomial approximation method is proposed in this section. The method treats transfer functions derived from truncated Y- Δ admittance from Alg. 2 to stabilize trans-

fer function approximations.

The problem of representing a high-order linear system by a reduced-order linear model is of considerable importance in simulation. One class of solutions, the so-called Padé approximation methods have been extremely successful in VLSI interconnect model reductions. AWE[6] and some of its variants are in this class. There is, however, one disadvantage of these, but not all, Padé methods. They may produce unstable reduced models from stable large-order models.

Due to Th. 6, reduced k -th order admittance resulted from Y- Δ transformation agrees with high-order admittance on the first k terms in both its numerator and denominator. So that a reduced-order transfer function derived from these reduced-order admittance also agrees with the corresponding high-order version on its first k terms.

But A truncated low-order transfer function of an originally passive linear system may not be stable, simply because a truncated low-order polynomial of a high-order Hurwitz polynomial may not be a Hurwitz polynomial any more. Since DTT method has no post-process on truncated admittance, it can not guarantee to generate stable transfer functions. We employ a method that approximates such transfer functions and the resultant approximants are guaranteed stable.

Consider a real rational function

$$H(s) = \frac{P(s)}{Q(s)} = \frac{1 + a_1s + \dots + a_ms^m}{b_0 + b_1s + \dots + b_ns^n}, \quad a_i, b_i > 0, \quad (19)$$

with the even and odd parts of $Q(s)$

$$\begin{aligned} m(s) &= b_0 + b_2s^2 + b_4s^4 + \dots + b_ns^n \\ n(s) &= b_1s + b_3s^3 + b_5s^5 + \dots + b_{n-1}s^{n-1} \end{aligned} \quad (20)$$

for n even, and with the last terms interchanged if the order n is odd. Here we suppose, merely to be specific, that the denominator of the given rational function is of even order. $Q(s)$ is said to be a Hurwitz polynomial if coefficients of quotient terms $q_0s^{-1}, q_1s^{-1}, \dots, q_{n-1}s^{-1}$ from continuous fraction of $n(s)/m(s)$ in (21) are all positive

$$\frac{n(s)}{m(s)} = \frac{1}{q_0s^{-1} + \frac{1}{q_1s^{-1} + \frac{1}{q_2s^{-1} + \frac{1}{q_3s^{-1} + \dots + \frac{1}{q_{n-1}s^{-1}}}}}}. \quad (21)$$

Actually the k -th ($k < n$) quotient in (21) depends only on the first $2k$ (or n , whichever is smaller) terms in (19). Thus if $Q(s)$ in (19) is a Hurwitz polynomial, the first k quotients of the denominator $Q_1(s)$ of a truncated $2k$ -th order polynomial of $Q(s)$ in

$$H_1(s) = \frac{P_1(s)}{Q_1(s)} = \frac{1 + a_1s + \dots + a_{2k}s^{2k}}{b_0 + b_1s + \dots + b_{2k}s^{2k}} \quad (22)$$

are the same as those of $Q(s)$ in (21). Therefore these k quotients are guaranteed positive.

Taking all the quotients of $Q_1(s)$ in (22) from the first up to and excluding the first negative one, we have a rational approximant of $m(s)/n(s)$ as

$$\frac{\tilde{n}(s)}{\tilde{m}(s)} \equiv \frac{1}{q_0 s^{-1} + \frac{1}{q_1 s^{-1} + \frac{1}{\dots + \frac{1}{q_{p-1} s^{-1}}}}}, \quad k \leq p \leq n \quad (23)$$

By definition, the polynomial

$$\tilde{Q}(s) = \tilde{m}(s) + \tilde{n}(s) \quad (24)$$

is a p -th order Hurwitz polynomial.

Because Y- Δ transformation helps preserve low-order terms of a high-order linear transfer function, we can perform a continuous fraction on the denominator of the reduced-order transfer function, and construct a Hurwitz characteristic polynomial $\tilde{Q}(s)$. $\tilde{Q}(s)$ is the denominator of our new transfer function

$$\tilde{H}(s) = \frac{\tilde{P}(s)}{\tilde{Q}(s)} \quad (25)$$

To evaluate the numerator $\tilde{P}(s)$ in (25), we match the time moments of $\tilde{H}(s)$ with those of $H(s)$ in (19) up to the k -th term, i.e.,

$$H(s) - \tilde{H}(s) = O(s^k). \quad (26)$$

Algorithm 3. Hurwitz Approximation

1. (Positive Quotients) For a given n -th order transfer function $H(s)$ from Alg. 2, do a continuous fraction on $H(s)$'s denominator. Save the first p positive quotients.
2. (Stable Denominator) Build a p -th order Hurwitz polynomial $\tilde{Q}(s)$, based on the p quotients.
3. (Numerator) Establish linear equations to evaluate the numerator $\tilde{P}(s)$ of $\tilde{H}(s)$ in (25), by matching the first p time moments of $\tilde{H}(s)$ with those of $H(s)$.

For example, given a real rational function

$$H(s) = \frac{P(s)}{Q(s)} = \frac{1 + 3s + 14s^2 + 25s^3 + 5s^4}{10 + 2s + 23s^2 + 4s^3 + 9s^4 + s^5}, \quad (27)$$

we can write $m(s)$ and $n(s)$ defined in (20) as

$$\begin{aligned} m(s) &= 10 + 23s^2 + 9s^4 \\ n(s) &= 2s + 4s^3 + s^5 \end{aligned}$$

We make a continuous fraction on $m(s)/n(s)$, then we have

$$\frac{n(s)}{m(s)} = \frac{1}{5s^{-1} + \frac{1}{\frac{2}{3}s^{-1} + \frac{1}{\frac{9}{4}s^{-1} + \frac{1}{\frac{16}{21}s^{-1} + \frac{1}{\frac{7}{4}s^{-1}}}}}} \quad (28)$$

Because all the coefficients of the quotients are positive, $P(s)$ in (27) is a Hurwitz polynomial. $H(s)$ is supposed to be a transfer function of a linear network.

If we apply a fourth order Y- Δ transformation on the network to approximant $H_1(s)$ of $H(s)$, we will get

$$H_1(s) = \frac{P_1(s)}{Q_1(s)} = \frac{1 + 3s + 14s^2 + 25s^3 + 5s^4}{10 + 2s + 23s^2 + 4s^3 + 9s^4},$$

we write $m_1(s)$ and $n_1(s)$ of $Q_1(s)$ as

$$\begin{aligned} m_1(s) &= 10 + 23s^2 + 9s^4 \\ n_1(s) &= 2s + 4s^3 \end{aligned}$$

And we make a similar continuous fraction on $m_1(s)/n_1(s)$

$$\frac{n_1(s)}{m_1(s)} = \frac{1}{5s^{-1} + \frac{1}{\frac{2}{3}s^{-1} + \frac{1}{-\frac{3}{2}s^{-1} + \frac{1}{-\frac{2}{9}s^{-1}}}}} \quad (29)$$

The first two quotients in (29) are positive and the same as those in (28), which is consistent with our conclusion above. The remaining two quotients have negative coefficients so that we simply ignore them. Following (24), we have the Hurwitz approximant

$$\tilde{Q}(s) = 10 + 2s + 3s^2. \quad (30)$$

As $m_0 = \frac{1}{10}$ and $m_1 = \frac{7}{25}$ are the first two moments of $H(s)$ in (27), it is straightforward to find out that $\tilde{P}(s) = 1 + 3s$ makes the first two moments of $\frac{\tilde{P}(s)}{\tilde{Q}(s)}$ equal to those of $H(s)$.

VIII. EXPERIMENTAL RESULTS

In this section, we show some results from the proposed generalized Y- Δ transformation. The experiments consist of three parts: 1) waveform evaluation, 2) pole analysis, and 3) impact of common-factor cancellations. A linear network simulation package was developed based on the proposed Y- Δ transformation reduction algorithms. Hurwitz approximation method was also implemented for waveform evaluation and pole analysis. Several industrial interconnect and power/ground circuits were used in our experiments. The CPU runtime was tested on a HP C3000 workstation.

Circuit	elements			CPU time(s)	
	#R	#L	#C	Y- Δ	spice3f4
Tree-like	1035	1034	1001	0.34	3.94
	16397	16394	14299	11.42	134
Mesh-like	1675	2439	733	5.22	73.19
	8305	0	8038	2.07	25.95
	66941	0	67119	41.25	1536.77

TABLE I
CPU RUNTIME USING Y- Δ TECHNIQUE AS COMPARED TO SPICE3F4 FOR FIVE INDUSTRIAL CIRCUITS.

Two tree-like circuits were tested. One was a set of uniform bus lines with resistance, inductance, fringe coupling capacitance and grounded capacitance. Inductance is in the magnitude of $10^{-11}(10pH)$ and capacitance is $10^{-14}(10fF)$. Rise time of input signal was $50ps$, and simulation time range was from 0 to $150ps$. The other was a clock distribution with similar RCL and simulation configurations. Y- Δ reduction was tested with 15 preserved order. Note that none of the two circuits was strictly tree structured.

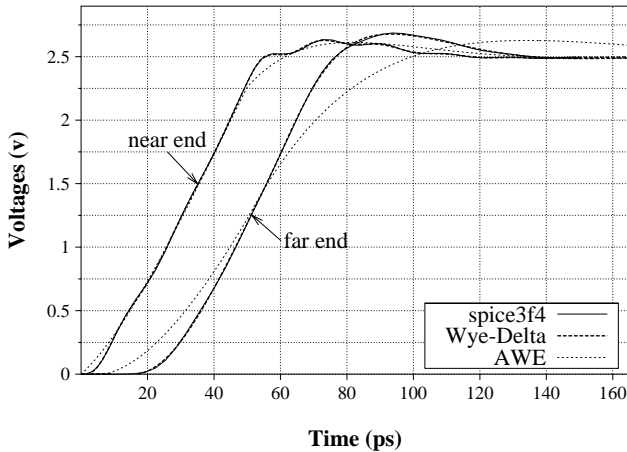


Fig. 4. Transient response evaluated using Y- Δ transformation with Hurwitz approximation as compared to AWE method and SPICE simulations for coupled RCL bus lines.

Fig. 4 shows transient response evaluated using 7-th Hurwitz approximants of admittance from Y- Δ transformation, as compared to AWE method and SPICE simulations for the two circuits. We only used 3rd order transfer functions from AWE method as higher order ones were not stable. Based on the circuits that we have, an average 10-fold improvement over SPICE in CPU runtime has been achieved with negligible errors from SPICE.

Fig. 5 plots locations of poles on complex plane. These poles are approximated for a mesh-like circuit in entry 4 of Table 1. A 30 order Y- Δ reduction was performed, and a

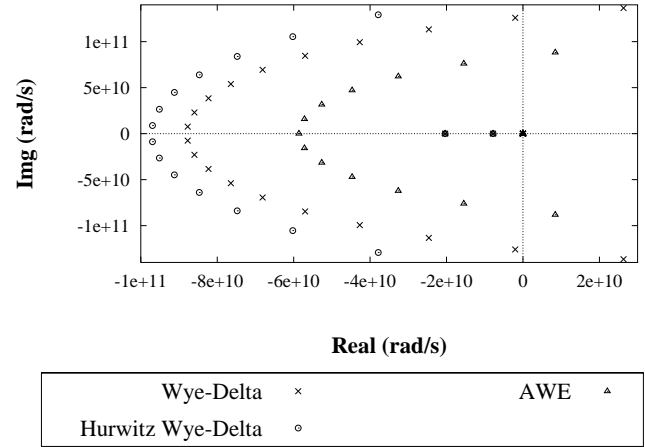


Fig. 5. Pole analysis using Y- Δ transformation with Hurwitz approximation as compared to Y- Δ transformation only and AWE method for RCL power/ground mesh.

16 order Hurwitz approximant was obtained, because in this test case, we happened to get one more positive quotient from the 30-order transfer function, which led to one more moment matched. Padé approximations and direct result of Y- Δ transformation had all resulted in positive real parts in some poles.

Fig. 6 shows the number of orders of admittance from our implementation of Y- Δ with consideration of common factor effects, compared to a naïve implementation without the consideration. The figure shows that the latter one grew expo-

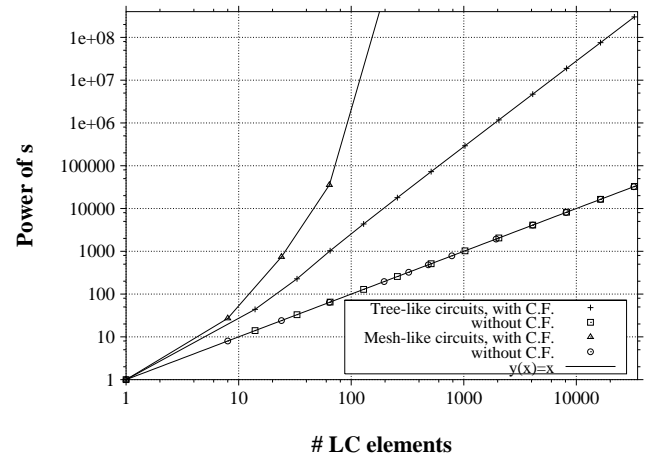


Fig. 6. Order of admittance after Y- Δ transformation with recognizing common factors as compared to a naïve implementation without recognizing common factors.

ponentially, for both tree-like circuits and mesh-like circuits. As we expected, orders of admittance in mesh-like circuits grew even faster than in tree-like circuits, as average degree of nodes

in the former cases was larger than that of nodes in the latter cases. Our implementation shows that when reduction completes with two nodes left, the order of the resultant admittance was the same as the order of original circuits, fitting the $y(x) = x$ line.

IX. CONCLUSIONS AND FUTURE WORK

We proposed a generalized Y- Δ transformation for linear network model reduction. The proposed algorithms can handle linear resistors, capacitors, self and mutual K elements and independent sources. The algorithm integrated common-factor-cancellation operations that were not seen in the literature. Admittance in reduced circuits has the guaranteed simplest form, without redundant common factors.

Realization of admittance from Y- Δ reduction could be achieved by formulating the problem in Geometric Programming. A set of simple low-order RCL circuits have been proposed as templates with unknown element values, by optimizing the number of admittance moments matched, the admittance could be realized into various reduced-order models.

X. ACKNOWLEDGMENT

This work was supported in part under grants from NSF project number MIP-9987678, the California MICRO program, and SRC support. Authors would like to thank the reviewers for their valuable suggestions.

REFERENCES

- [1] T.H. Chen, C.P. Chen, "Efficient large-scale power grid analysis based on preconditioned Krylov-subspace iterative methods," in *DAC*, pp.559–62, 2000.
- [2] J.W. Demmel, J.R. Gulbert, X.S. Li, "SuperLU user's guide," *National Energy Research Scientific Computing Center*, <http://www.nersc.gov/xiaoye/SuperLU>, 1999
- [3] Z. Qin, Z. Zhu, and C.K. Cheng, "Efficient transient analysis for large linear networks," in *SASIMI*, pp.293–00, 2001.
- [4] S. R. Nassif, J. N. Kozhaya, "Fast power grid simulation," in *DAC*, pp.156–61, 2000.
- [5] J. N. Kozhaya, S. R. Nassif, and F. N. Najm, "Multigrid-like technique for power grid analysis," in *ICCAD*, pp.480–7, 2001.
- [6] L.T. Pillage and R.A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. on CAD*, vol. CAD-9, pp.352–66, Apr. 1990.
- [7] S. Lin, and E. S. Kuh, "transient simulation of lossy interconnects based on the recursive convolution formulation," *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications*, vol.39, pp.879–92, Nov. 1992.
- [8] H. Liao, W. Dai, R. Wang, and F. Y. Chang, "S-parameter based macro model of distributed-lumped networks using exponentially decayed polynomial function," in *DAC*, pp.726–31, 1993.
- [9] C. L. Ratzlaff, L. T. Pillage, "RICE: rapid interconnect circuit evaluation using AWE," *IEEE Trans. on CAD*, vol 13, pp.763–6, Jun. 1994.
- [10] M. Sriram, S. M. Kahng, "fast approximation of the transient response of lossy transmission line trees," in *DAC*, pp.691–6, 1993.
- [11] M. Sriram, S. M. Kahng, "Performance driven MCM routing using a second order RLC tree delay model," *Proceedings. fifth Annual IEEE International Conference on Wafer Scale Integration*, pp.262–7, 1993.
- [12] H. Liao, W. W.M. Dai, "Partitioning and reduction of RC interconnect networks based on scattering parameter macromodels," in *DAC*, pp.704–9, 1995.
- [13] P. Feldmann, R. W. Freund, "Reduced-order modeling of large linear subcircuits via a block Lanczos algorithm," in *DAC*, pp.376–80, 1995.
- [14] D. L. Boley, "Krylov space methods on state-space control models," in *Circuits Systems and Signal Processing*, vol. 13, no.6, pp.733–58, 1994.
- [15] A. Odabasioglu, M. Celik, L. T. Pillage, "PRIMA: passive reduced-order interconnect macromodeling algorithm," *IEEE Trans. on CAD*, vol.17, pp.645–54, Aug. 1998.
- [16] K. J. Kerns, I. L. Wemple, A. T. Yang, "Stable and efficient reduction of substrate model networks using congruence transforms," in *ICCAD*, pp. 207–14, 1995.
- [17] K. J. Kerns, "Accurate and stable reduction of RLC networks using split congruence transformations," Ph.D. Dissertation, Univ. Washington, Sept. 1996.
- [18] S. Chan, "Introductory topological analysis of electrical networks," *Holt, Rinehart and Winston, Inc.*, 1974.
- [19] Y. I. Ismail, E. G. Friedman, "DTT: direct truncation of the transfer function — an alternative to moment matching for tree structured interconnect," *IEEE Trans. on CAD of ICAS*, vol. 21, No. 2, Feb. 2002.
- [20] S. B. Akers, Jr., "The use of wye-delta transformations in network simplification," *Operations Research*, 8, pp.311-23, 1960.
- [21] A. Devgan, H. Ji, W. Dai, "How to efficiently capture on-chip inductance effects: introducing a new circuit element K ," in *ICCAD*, pp.150–5, 2000.
- [22] H. Ji, A. Devgan, W. Dai, "KSim: a stable and efficient RKC simulator for capturing on-chip inductance effect," in *ASP-DAC*, pp.379–84, 2001.
- [23] A. George, J. W.H. Liu, "computer solution of large sparse positive definite systems," *Prentice-Hall*, Englewood Cliffs, NJ, 1981.
- [24] J. W.H. Liu, "modification of the minimum-degree algorithm by multiple elimination," *ACM Trans. Math Software*, pp.337–58, vol.11, issue 5, Jun. 1985.
- [25] M. Yannakakis, "computing the minimum fill-in is NP-complete," *SIAM J. Alg. Disc. Math*, 1, pp.77–9, 1981.
- [26] Z. Qin, C. Cheng, "linear network reduction via generalized Y- Δ transformation: theory," *Technical Report CS2002–0706*, May, 2002.