PRIMA: Passive Reduced-Order Interconnect Macromodeling Algorithm

Altan Odabasioglu, Mustafa Celik, *Member, IEEE*, and Lawrence T. Pileggi, *Senior Member, IEEE*

***Abstract*— This paper describes an algorithm for generating provably passive reduced-order N-port models for *RLC* interconnect circuits. It is demonstrated that, in addition to macromodel stability, macromodel passivity is needed to guarantee the overall circuit stability once the active and passive driver/load models are connected. The approach proposed here, PRIMA, is a general method for obtaining passive reduced-order macromodels for linear *RLC* systems. In this paper, PRIMA is demonstrated in terms of a simple implementation which extends the block Arnoldi technique to include guaranteed passivity while providing superior accuracy. While the same passivity extension is not possible for MPVL, comparable accuracy in the frequency domain for all examples is observed.**

I. Introduction

As integrated circuits and systems continue to be designed smaller and for faster operation, *RLC* interconnect effects have a more dominant impact on signal propagation than ever before. In addition, parasitic coupling effects and reduced power supply voltage levels make interconnect modeling increasingly important. Since these interconnect models can contain thousands of tightly coupled *R–L–C* components, reduced-order macromodels are imperative [1]–[4]. Ideally, a simulator would isolate the large linear portions of the circuit from the nonlinear elements (e.g., transistor models) and preprocess them into reduced order multiport macromodels (Fig. 1).

It is well known that an N port can be fully represented by its admittance parameters in the Laplace domain (Fig. 2); however, the objective is to apply model order reduction to produce low-order rational approximations for each entry in the admittance matrix, $\boldsymbol{Y}(s)$. A single-input single-output (SISO) N-port model approach would perform model order reduction on each term Y_{ij} individually. Both asymptotic waveform evaluation (AWE) [1] and Padé via Lanczos (PVL) [2], which are Padé approximations, can perform SISO reduction by matching $2q$ moments for a qth-order approximation of each Y_{ij} term. The Arnoldi algorithm [4] can also be used to obtain SISO approximations; however, it matches only q moments for a qth-order approximation. MPVL (matrix Padé via Lanczos) [5] and block Arnoldi [6] are multi-input multi-output (MIMO) versions of PVL and Arnoldi, respectively. The projection perspective to the block Lanczos process has also been developed [7]. In the block techniques, the system modified nodal analysis (MNA) matrices are directly reduced by matrix transformations. Refer to [8] for a recent summary of useful order reduction methods.

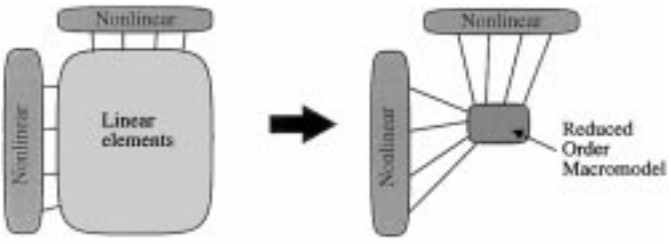

Fig. 1. Preprocessing of linear portion of circuit into N-port macromodels.

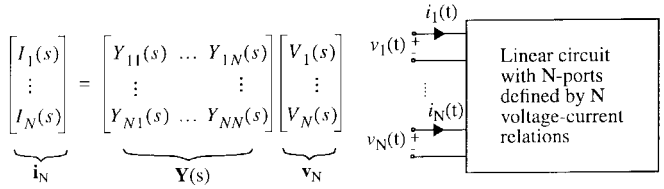

Fig. 2. Multiport representation of a linear circuit.

In general, the reduced-order model of an *RLC* circuit can have unstable poles. Although it is always possible to obtain an asymptotically stable model by simply discarding the unstable poles, passivity is not guaranteed. The coordinate-transformed Arnoldi algorithm proposed in [9] was introduced as a remedy for the instability problem, but it cannot guarantee passivity. The PACT algorithm [3] proposed a new direction for the passive reduced-order model for *RC* circuits based on congruence transformations. More recently, the same authors proposed split congruence transformations [10] for passive reductions of *RLC* circuits, producing equivalent circuit realizations. In [10], however, the extra steps required to split the transformation matrix can result in a decrease in accuracy.

A passive system denotes a system that is incapable of generating energy, and hence one that can only absorb energy from the sources used to excite it [11]. Passivity is an important property to satisfy because stable, but not passive macromodels can produce unstable systems when connected to other stable, even passive, loads. A property in classical circuit theory states that: interconnections of stable systems may not necessarily be stable; but (strictly) passive circuits are (asymptotically) stable;

Manuscript received February 6, 1997. This work was supported in part by the Defense Advanced Research Projects Agency, sponsored by the Air Force Office of Scientific Research, under Grant F49620-96-1-0069, and by grants from Intel Corporation and SGS-Thomson Microelectronics. This paper was recommended by Associate Editor K. Mayaram.

A. Odabasioglu is with Monterey Design Systems, Sunnyvale, CA 94089 USA.

M. Celik is with Pacific Numerix Corporation, Scottsdale, AZ 85258 USA.

L. T. Pileggi is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA.

Publisher Item Identifier S 0278-0070(98)05822-9.

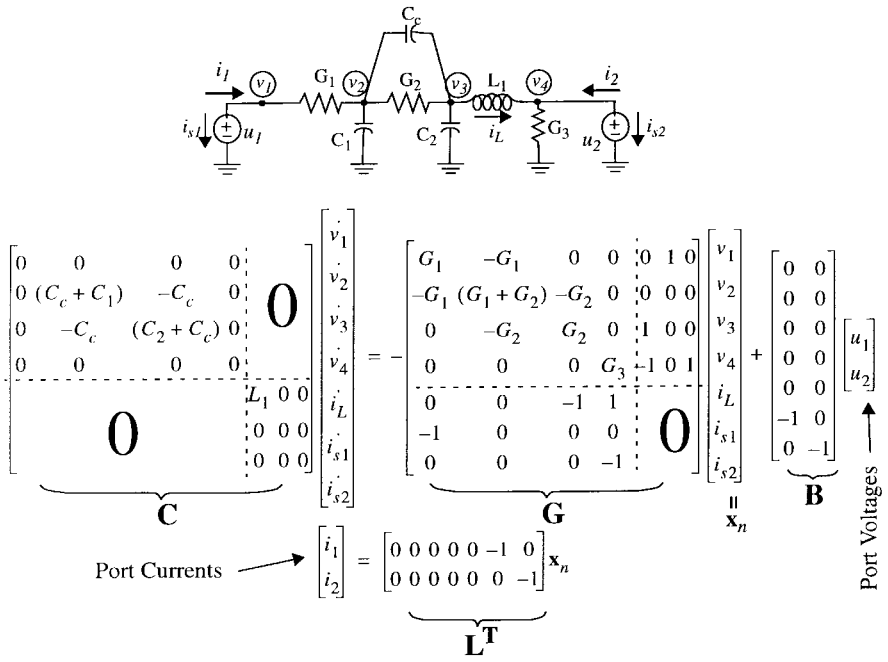

Fig. 3. Illustration of the formation of the MNA matrices for a two-port *RLC* circuit example.

and arbitrary interconnections of (strictly) passive circuits are (strictly) passive, and therefore, (asymptotically) stable [12].

In this paper, we propose a passive reduced-order interconnect macromodeling algorithm, PRIMA, based on the block Arnoldi algorithm, but with congruence transformations that produce provably passive reduced-order macromodels for arbitrary *RLC* circuits. PRIMA demonstrates accuracy comparable to MPVL and superior to block Arnoldi. The underlying technique for passive reduction, which is the basis for PRIMA, is also general enough to employ numerous Krylov space generation methods [7], [8], [13].

II. Background

As shown in Fig. 3, voltage sources are connected to the ports to obtain the admittance matrix of a multiport. The N port, along with these sources, constitutes our time-domain modified nodal analysis (MNA) circuit equations:

$$\begin{aligned} \boldsymbol{C}\dot{\boldsymbol{x}}_n &= -\boldsymbol{G}\boldsymbol{x}_n + \boldsymbol{B}\boldsymbol{u}_N \\ \boldsymbol{i}_N &= \boldsymbol{L}^T\boldsymbol{x}_n. \end{aligned} \tag{1}$$

The $\boldsymbol{i}_N$ and $\boldsymbol{u}_N$ vectors denote the port currents and voltages, respectively, and

$$\boldsymbol{C} \equiv \begin{bmatrix} \boldsymbol{Q} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{H} \end{bmatrix} \quad \boldsymbol{G} \equiv \begin{bmatrix} \boldsymbol{N} & \boldsymbol{E} \\ -\boldsymbol{E}^T & \boldsymbol{0} \end{bmatrix} \quad \boldsymbol{x}_n \equiv \begin{bmatrix} \boldsymbol{v} \\ \boldsymbol{i} \end{bmatrix} \tag{2}$$

where $\boldsymbol{v}$ and $\boldsymbol{i}$ are the MNA variables [yielding a total number of n unknowns in (1)] corresponding to the node voltages and the branch currents for voltage sources and inductors, respectively. The matrices $\boldsymbol{G} \in \Re^{n\times n}$ and $\boldsymbol{C} \in \Re^{n\times n}$ represent the conductance and susceptance matrices (except that the rows corresponding to the current variables are negated as in [9]). $\boldsymbol{N}$, $\boldsymbol{Q}$, and $\boldsymbol{H}$ are the matrices containing the stamps for resistors, capacitors, and inductors, respectively. $\boldsymbol{E}$ consists of ones, minus ones, and zeros, which represent the current variables in KCL equations. Provided that the original N port is composed of passive linear elements only, $\boldsymbol{Q}$, $\boldsymbol{H}$, and $\boldsymbol{N}$ are symmetric nonnegative definite matrices. It is straightforward to show that $\boldsymbol{C}$ is a symmetric nonnegative definite matrix with this MNA formulation.

An MNA formulation example for a small circuit is shown in Fig. 3. Since this is an N-port formulation, whereby the only sources are the voltage sources at the N-port nodes, $\boldsymbol{B} = \boldsymbol{L}$ where $\boldsymbol{B} \in \Re^{n\times N}$. But we will maintain the separate $\boldsymbol{B}$ and $\boldsymbol{L}$ notation throughout this paper for equation generality.

Assuming for now that we are interested in admittance parameters and returning to (1), following the notation in [2], we define

$$\boldsymbol{A} \equiv -\boldsymbol{G}^{-1}\boldsymbol{C} \quad \text{and} \quad \boldsymbol{R} \equiv \boldsymbol{G}^{-1}\boldsymbol{B}. \tag{3}$$

With unit impulse voltages at the ports, taking the Laplace transformation of (1) and solving for the port current variables, the y-parameter matrix is given as

$$\boldsymbol{Y}(s) = \boldsymbol{L}^T(\boldsymbol{G} + s\boldsymbol{C})^{-1}\boldsymbol{B}. \tag{4}$$

Using (3), the admittance matrix can also be expressed as

$$\boldsymbol{Y}(s) = \boldsymbol{L}^T(\boldsymbol{I}_n - s\boldsymbol{A})^{-1}\boldsymbol{R} \tag{5}$$

where $\boldsymbol{I}_n$ is the $n \times n$ identity matrix. It is apparent from (5) that the eigenvalues of $\boldsymbol{A}$ represent the reciprocal poles of $\boldsymbol{Y}(s)$. We can define the impedance parameter matrix $\boldsymbol{Z}(s)$ in a similar way. Specifically, with the unit impulse current sources connected to the N port, the port voltages would represent the z-parameter terms.

Considering either the admittance or impedance representation, we define the block moments and the block Krylov space as follows for this paper.

Definition 1: The *block moments* of $\boldsymbol{Y}(s)$ are defined as the coefficients of Taylor expansion of $\boldsymbol{Y}(s)$ around $s = 0$:

$$\boldsymbol{Y}(s) = \boldsymbol{M}_0 + \boldsymbol{M}_1 s + \boldsymbol{M}_2 s^2 + \cdots \tag{6}$$

where $\boldsymbol{M}_i \in \Re^{N\times N}$. These block moments can be computed using the relation

$$\boldsymbol{M}_i = \boldsymbol{L}^T \boldsymbol{A}^i \boldsymbol{R}. \tag{7}$$

In circuit terms, the entry in the jth row, kth column of $\boldsymbol{M}_i$ is the ith moment of the current that flows into port j when the voltage source at port k is the only nonzero source.

Definition 2: The *block Krylov space* generated by matrices $\boldsymbol{A} \in \Re^{n\times n}$ and $\boldsymbol{R} = [\boldsymbol{r}_0\ \boldsymbol{r}_1 \cdots \boldsymbol{r}_N] \in \Re^{n\times N}$ are defined as[1]

$$\begin{aligned} Kr(\boldsymbol{A}, \boldsymbol{R}, q) \equiv\ & colsp\big[\boldsymbol{R}, \boldsymbol{AR}, \boldsymbol{A}^2\boldsymbol{R}, \cdots, \\ & \boldsymbol{A}^{k-1}\boldsymbol{R}, \boldsymbol{A}^k\boldsymbol{r}_0, \boldsymbol{A}^k\boldsymbol{r}_1, \cdots, \boldsymbol{A}^k\boldsymbol{r}_l\big] \\ & k = \lfloor q/N \rfloor, \quad l = q - kN. \end{aligned} \tag{8}$$

From the circuit's perspective, the vector $\boldsymbol{A}^j\boldsymbol{r}_k$ contains the set of jth moments for the MNA variables [x in (2)] when the kth source is active and all other sources in the circuit are set to be zero. In matrix terms, the block Krylov space spans the combination of moment vectors generated by different sources in the circuit.

A. Block Arnoldi Algorithm

The block Arnoldi algorithm reduces the system matrix $\boldsymbol{A}$ in (3) to a small block upper Hessenberg matrix $\boldsymbol{H}_q$. The algorithm involves successively filling in the columns of $\boldsymbol{X}$ in the relation $\boldsymbol{AX} = \boldsymbol{XH}_q$ subject to $\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{I}_q$. Here, $\boldsymbol{X} \in \Re^{n\times q}$ is an orthonormal matrix spanning the Krylov space $Kr(\boldsymbol{A}, \boldsymbol{R}, q)$, $\boldsymbol{H}_q \in \Re^{q\times q}$ is a block upper Hessenberg matrix,[2] and $\boldsymbol{I}_q \in \Re^{q\times q}$ is an identity matrix. In summary

$$\begin{aligned} colsp(\boldsymbol{X}) &= Kr(\boldsymbol{A}, \boldsymbol{R}, q) \\ \boldsymbol{X}^T\boldsymbol{AX} &= \boldsymbol{H}_q \\ \boldsymbol{X}^T\boldsymbol{X} &= \boldsymbol{I}_q. \end{aligned} \tag{9}$$

Finding the reduced-order admittance matrix can be explained by a change of variable in (1):

$$\boldsymbol{x}_n = \boldsymbol{X}\boldsymbol{z}_q \tag{10}$$

where $\boldsymbol{z}_q \in \Re^{q\times 1}$ is now the reduced-order system variable. This reduces the number of unknowns in the system since q is generally much smaller than n. Substituting (10) into (1), then premultiplying first by $\boldsymbol{G}^{-1}$ and then by $\boldsymbol{X}^T$ yields

$$\begin{aligned} -\boldsymbol{X}^T\boldsymbol{G}^{-1}\boldsymbol{CX}\dot{\boldsymbol{z}}_q &= \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{z}_q - \boldsymbol{X}^T\boldsymbol{G}^{-1}\boldsymbol{B}\boldsymbol{u}_N \\ \boldsymbol{i}_N &= \boldsymbol{L}^T\boldsymbol{X}\boldsymbol{z}_q. \end{aligned} \tag{11}$$

Recalling (3) and using the relations in (9) gives

$$\begin{aligned} \boldsymbol{H}_q\dot{\boldsymbol{z}}_q &= \boldsymbol{z}_q - \boldsymbol{X}^T\boldsymbol{R}\boldsymbol{u}_N \\ \boldsymbol{i}_N &= \boldsymbol{L}^T\boldsymbol{X}\boldsymbol{z}_q. \end{aligned} \tag{12}$$

[1] The $\lfloor\cdot\rfloor$ operator is the truncation to the nearest integer toward zero.

[2] A matrix $\boldsymbol{H}$ is an upper Hessenberg matrix if $\boldsymbol{H}_{ij} = 0$ when $i > j + 1$.

Therefore, in the Laplace domain

$$\hat{\boldsymbol{Y}}(s) = \boldsymbol{L}^T\boldsymbol{X}(\boldsymbol{I}_q - s\boldsymbol{H}_q)^{-1}\boldsymbol{X}^T\boldsymbol{R}. \tag{13}$$

The reduced-order system equations and admittance matrix are given by (12) and (13), respectively. The poles of the reduced-order system are the reciprocal eigenvalues of $\boldsymbol{H}_q$. A complete pole/residue decomposition can be obtained by eigendecomposing $\boldsymbol{H}_q$

$$\boldsymbol{H}_q = \boldsymbol{S}\Lambda_q\boldsymbol{S}^{-1} \tag{14}$$

$$\hat{\boldsymbol{Y}}(s) = \boldsymbol{L}^T\boldsymbol{XS}(\boldsymbol{I}_q - s\Lambda_q)^{-1}\boldsymbol{S}^{-1}\boldsymbol{X}^T\boldsymbol{R}. \tag{15}$$

The inversion of $(\boldsymbol{I}_q - s\Lambda_q)$ is trivial because it is a diagonal matrix.

In [6], it is shown that

$$\boldsymbol{A}^i\boldsymbol{R} = \boldsymbol{X}\boldsymbol{H}_q^i\boldsymbol{X}^T\boldsymbol{R}, \qquad 0 \le i < \left\lfloor \frac{q}{N} \right\rfloor \tag{16}$$

from which one can derive that the first $\lfloor q/N \rfloor$ block moments of $\hat{\boldsymbol{Y}}(s)$ in (13) and $\boldsymbol{Y}(s)$ in (5) match. In fact, it can be shown that if $\boldsymbol{X}$ spans the kth moment vector that is generated by the jth source, then the kth columns of the jth block moments of $\hat{\boldsymbol{Y}}(s)$ and $\boldsymbol{Y}(s)$ match.

The accuracy of the block Arnoldi approximation gradually increases as the order is increased since more moments of the original admittance matrix will be matched. Since it does not directly use moments, the block Arnoldi algorithm does not suffer from the same numerical inaccuracy as AWE. The algorithm requires one LU decomposition of $\boldsymbol{G}$ and q backward–forward substitutions to generate the block Krylov space.

III. PRIMA: Passive Reduced-Order Interconnect Macromodeling Algorithm

In this section, we present our passive reduced-order macromodeling algorithm. The algorithm given here is based on the block Arnoldi algorithm. However, it should be noted that PRIMA is a general technique for the passive reduction of *RLC* circuits, and is not bound to a particular Arnoldi or Lanczos process. The PRIMA algorithm, as applied with the block Arnoldi algorithm, is summarized in Fig. 4.

After $\lfloor q/N \rfloor + 1$ (the extra step is not necessary when (q/N) is an integer) iterations of the block Arnoldi algorithm, the $n \times q$ matrix $\boldsymbol{X}$ and a $q \times q$ upper Hessenberg matrix $\boldsymbol{H}$ are found such that

$$\begin{aligned} colsp(\boldsymbol{X}) &= Kr(\boldsymbol{A}, \boldsymbol{R}, q) \\ \boldsymbol{X}^T\boldsymbol{X} &= \boldsymbol{I}_q \\ \boldsymbol{X}^T\boldsymbol{AX} &= \boldsymbol{H}. \end{aligned} \tag{17}$$

In the classical Arnoldi approach described in Section II-A and as employed in [4], the reduced-order $\boldsymbol{Y}(s)$ is given as

$$\hat{\boldsymbol{Y}}(s) = \boldsymbol{L}^T\boldsymbol{X}(\boldsymbol{I} - s\boldsymbol{H})^{-1}\boldsymbol{X}^T\boldsymbol{R}. \tag{18}$$

In the PRIMA algorithm, the conductance and susceptance matrices are directly reduced so that passivity is preserved

❍Connect voltage sources to the multiport & obtain the MNA matrices as in (2).

❍Set $[\,\mathbf{b}_1 \mid \mathbf{b}_2 \mid \ldots \mid \mathbf{b}_N\,] = \mathbf{B}$ and $[\,\mathbf{l}_1 \mid \mathbf{l}_2 \mid \ldots \mid \mathbf{l}_N\,] = \mathbf{L}$

❍Solve $\mathbf{G}\mathbf{R} = \mathbf{B}$ for $\mathbf{R}$

❍$(\mathbf{X}_0, \mathbf{T}) = qr(\mathbf{R})$; qr factorization of $\mathbf{R}$

❍If $\frac{q}{N}$ is not an integer, set $n = \left\lfloor \frac{q}{N} \right\rfloor + 1$, else set $n = \frac{q}{N}$

❍For k=1, 2, ..., n

Set $\mathbf{V} = \mathbf{C}\mathbf{X}_{k-1}$

Solve $\mathbf{G}\,\mathbf{X}_k^{(0)} = \mathbf{V}$ for $\mathbf{X}_k^{(0)}$

For j=1, .., k

$\mathbf{H} = \mathbf{X}_{k-j}^T \mathbf{X}_k^{(j-1)}$

$\mathbf{X}_k^{(j)} = \mathbf{X}_k^{(j-1)} - \mathbf{X}_{k-j}\,\mathbf{H}$

$(\mathbf{X}_k, \mathbf{T}) = qr(\mathbf{X}_k^{(k)})$; qr factorization of $\mathbf{X}_k^{(k)}$

❍Set $\mathbf{X} = \left[\mathbf{X}_0\ \mathbf{X}_1\ \ldots\ \mathbf{X}_{k-1}\right]$ and truncate $\mathbf{X}$ so that it has q columns only.

❍Compute $\tilde{\mathbf{C}} = \mathbf{X}^T\mathbf{C}\,\mathbf{X}$, $\tilde{\mathbf{G}} = \mathbf{X}^T\mathbf{G}\,\mathbf{X}$

❍Realize the macromodel as outlined in Section 4.1 or

❍Find eigendecomposition of $\tilde{\mathbf{G}}^{-1}\tilde{\mathbf{C}}$: $\tilde{\mathbf{G}}^{-1}\tilde{\mathbf{C}} = \mathbf{S}\,\Lambda\,\mathbf{S}^{-1}$

❍$\Lambda = diag(\lambda_1, \lambda_2, \ldots, \lambda_q)$

❍To find poles and residues for $Y_{i,j}(s)$:

Solve $\tilde{\mathbf{G}}\,\mathbf{w} = \mathbf{X}^T\mathbf{b}_j$ for $\mathbf{w}$

Set $\mu = \mathbf{S}^T\mathbf{X}^T\mathbf{l}_i$ and $\nu = \mathbf{S}^{-1}\mathbf{w}$

$$Y_{i,j}(s) = \sum_{n=1}^{q} \frac{\mu_n \nu_n}{1 - s\lambda_n}$$

❍Compute all $Y_{i,j}(s)$ to find $\mathbf{Y}(s) = \begin{bmatrix} Y_{1,1} & \cdots & Y_{1,N} \\ \vdots & & \vdots \\ Y_{N,1} & \cdots & Y_{N,N} \end{bmatrix}$

Fig. 4. Simple PRIMA implementation.

during reduction. Applying the change of variable $\boldsymbol{x}_n = \boldsymbol{X}\tilde{\boldsymbol{x}}_q$ in (1), and multiplying the first row by $\boldsymbol{X}^T$ from (17) yields

$$\begin{aligned}(\boldsymbol{X}^T\boldsymbol{C}\boldsymbol{X})\dot{\tilde{\boldsymbol{x}}}_q &= -(\boldsymbol{X}^T\boldsymbol{G}\boldsymbol{X})\tilde{\boldsymbol{x}}_q + (\boldsymbol{X}^T\boldsymbol{B})\boldsymbol{u}_N \\ \boldsymbol{i}_N &= (\boldsymbol{L}^T\boldsymbol{X})\tilde{\boldsymbol{x}}_q. \end{aligned} \tag{19}$$

So, for the macromodel, the reduced-order MNA matrices are

$$\begin{aligned}\tilde{\boldsymbol{C}} &= \boldsymbol{X}^T\boldsymbol{C}\boldsymbol{X} \quad \tilde{\boldsymbol{G}} = \boldsymbol{X}^T\boldsymbol{G}\boldsymbol{X} \\ \tilde{\boldsymbol{B}} &= \boldsymbol{X}^T\boldsymbol{B} \quad\ \ \tilde{\boldsymbol{L}} = \boldsymbol{X}^T\boldsymbol{L}. \end{aligned} \tag{20}$$

These types of transformations are known as congruence transformations. Congruence transformations were first introduced by [3] for order reduction of circuits. From (19) and (20), the reduced $\boldsymbol{Y}(s)$, namely, $\hat{\boldsymbol{Y}}(s)$, is now

$$\hat{\boldsymbol{Y}}(s) = \tilde{\boldsymbol{L}}^T\left(\tilde{\boldsymbol{G}} + s\tilde{\boldsymbol{C}}\right)^{-1}\tilde{\boldsymbol{B}}. \tag{21}$$

Since the size of $\tilde{\boldsymbol{G}}$ and $\tilde{\boldsymbol{C}}$ is typically very small, it is easy to find the poles and zeros of $\hat{\boldsymbol{Y}}(s)$ by eigendecomposition. The complexity of the algorithm is basically equivalent to that of the block Arnoldi process. As noticed, the block Arnoldi algorithm is used only as a means to generate the block Krylov space $\boldsymbol{X}$ that is used for the congruence transformations. We will further explore the connections between PRIMA and block Arnoldi in Section III-C.

A. Preservation of Passivity

If the system described by (1) and (2) is reduced by the transformations in (20), it can be shown that the reduced system is always passive. In [14], necessary and sufficient conditions for the system admittance matrix $\hat{\boldsymbol{Y}}(s) = \tilde{\boldsymbol{L}}^T(\tilde{\boldsymbol{G}} + s\tilde{\boldsymbol{C}})^{-1}\tilde{\boldsymbol{B}}$ to be passive are the following.

1) $\hat{\boldsymbol{Y}}(s^*) = \hat{\boldsymbol{Y}}^*(s)$ for all complex s, where $*$ is the complex conjugate operator.
2) $\hat{\boldsymbol{Y}}(s)$ is a positive matrix, that is, $\boldsymbol{z}^{*T}(\hat{\boldsymbol{Y}}(s) + \hat{\boldsymbol{Y}}^T(s^*))\boldsymbol{z} \geq 0$ for all complex values of s satisfying $\mathrm{Re}(s) > 0$ and for any complex vector $\boldsymbol{z}$.

The second condition also implies the analyticity of $\hat{\boldsymbol{Y}}(s)$ for $\mathrm{Re}(s) > 0$ since $\hat{\boldsymbol{Y}}(s)$ is a rational function of s (details in [14]). Therefore, the test of analyticity is unnecessary.

Due to the fact that the reduced matrices $\tilde{\boldsymbol{G}}, \tilde{\boldsymbol{C}}, \tilde{\boldsymbol{B}}$, and $\tilde{\boldsymbol{L}}$ are all real since the transformation matrix $\boldsymbol{X}$ is real, condition 1) is automatically satisfied. To show that condition 2) is satisfied, we first set $\boldsymbol{Y}_h(s) = \hat{\boldsymbol{Y}}(s) + \hat{\boldsymbol{Y}}^T(s^*)$ and use the property $\tilde{\boldsymbol{B}} = \tilde{\boldsymbol{L}}$ (since $\boldsymbol{B} = \boldsymbol{L}$ in our formulation, $\boldsymbol{X}^T\boldsymbol{B} = \boldsymbol{X}^T\boldsymbol{L}$) and some algebra to obtain

$$\begin{aligned}&\boldsymbol{z}^{*T}\boldsymbol{Y}_h(s)\boldsymbol{z} \\ &\quad= \boldsymbol{z}^{*T}\left(\tilde{\boldsymbol{B}}^T\left(\tilde{\boldsymbol{G}} + s\tilde{\boldsymbol{C}}\right)^{-1}\tilde{\boldsymbol{B}} + \tilde{\boldsymbol{B}}^T\left(\tilde{\boldsymbol{G}} + s^*\tilde{\boldsymbol{C}}\right)^{-T}\tilde{\boldsymbol{B}}\right)\boldsymbol{z} \\ &\quad= \boldsymbol{z}^{*T}\tilde{\boldsymbol{B}}^T\left[\left(\tilde{\boldsymbol{G}} + s\tilde{\boldsymbol{C}}\right)^{-1} + \left(\tilde{\boldsymbol{G}} + s^*\tilde{\boldsymbol{C}}\right)^{-T}\right]\tilde{\boldsymbol{B}}\boldsymbol{z} \\ &\quad= \boldsymbol{z}^{*T}\tilde{\boldsymbol{B}}^T\left(\tilde{\boldsymbol{G}} + s\tilde{\boldsymbol{C}}\right)^{-1}\left[\left(\tilde{\boldsymbol{G}} + s\tilde{\boldsymbol{C}}\right) + \left(\tilde{\boldsymbol{G}} + s^*\tilde{\boldsymbol{C}}\right)^{T}\right] \\ &\qquad\times \left(\tilde{\boldsymbol{G}} + s^*\tilde{\boldsymbol{C}}\right)^{-T}\tilde{\boldsymbol{B}}\boldsymbol{z}. \end{aligned} \tag{22}$$

Setting $\boldsymbol{w} = (\tilde{\boldsymbol{G}} + s^*\tilde{\boldsymbol{C}})^{-T}\tilde{\boldsymbol{B}}\boldsymbol{z}$ and $s = j\omega + \sigma$ yields

$$\begin{aligned}&\boldsymbol{z}^{*T}\boldsymbol{Y}_h(s)\boldsymbol{z} \\ &\quad= \boldsymbol{w}^{*T}\left[\left(\tilde{\boldsymbol{G}} + (\sigma + j\omega)\tilde{\boldsymbol{C}}\right) + \left(\tilde{\boldsymbol{G}} + (\sigma - j\omega)\tilde{\boldsymbol{C}}\right)^{T}\right]\boldsymbol{w} \\ &\quad= \boldsymbol{w}^{*T}\left[\tilde{\boldsymbol{G}} + \tilde{\boldsymbol{G}}^T + \sigma\left(\tilde{\boldsymbol{C}} + \tilde{\boldsymbol{C}}^T\right)\right]\boldsymbol{w} \\ &\quad= \boldsymbol{w}^{*T}\left[\boldsymbol{X}^T\boldsymbol{G}\boldsymbol{X} + \boldsymbol{X}^T\boldsymbol{G}^T\boldsymbol{X} + \sigma\left(\boldsymbol{X}^T\boldsymbol{C}\boldsymbol{X} + \boldsymbol{X}^T\boldsymbol{C}^T\boldsymbol{X}\right)\right]\boldsymbol{w} \\ &\quad= \boldsymbol{w}^{*T}\boldsymbol{X}^T\left[\boldsymbol{G} + \boldsymbol{G}^T + \sigma\left(\boldsymbol{C} + \boldsymbol{C}^T\right)\right]\boldsymbol{X}\boldsymbol{w}. \end{aligned} \tag{23}$$

Similarly, let $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}$ to get

$$\boldsymbol{z}^{*T}\boldsymbol{Y}_h(s)\boldsymbol{z} = \boldsymbol{y}^{*T}\left[\boldsymbol{G} + \boldsymbol{G}^T + \sigma\left(\boldsymbol{C} + \boldsymbol{C}^T\right)\right]\boldsymbol{y}. \tag{24}$$

Since $\boldsymbol{C}$ is symmetric, $\boldsymbol{C}^T + \boldsymbol{C} = 2\boldsymbol{C}$. $\boldsymbol{C}$ is known to be nonnegative definite [since we negate the rows corresponding to current variables as in (2)] so

$$\boldsymbol{y}^{*T}\sigma(\boldsymbol{C}^T + \boldsymbol{C})\boldsymbol{y} = 2\sigma\boldsymbol{y}^{*T}\boldsymbol{C}\boldsymbol{y} \geq 0 \tag{25}$$

for any complex vector $\boldsymbol{y}$ and $\sigma = \mathrm{Re}(s) > 0$. $\boldsymbol{N}$ (the resistor stamps) is a symmetric nonnegative definite matrix; therefore

$$\begin{aligned}\boldsymbol{y}^{*T}(\boldsymbol{G}^T + \boldsymbol{G})\boldsymbol{y} &= \boldsymbol{y}^{*T}\left(\begin{bmatrix}\boldsymbol{N} & \boldsymbol{E}\\ -\boldsymbol{E}^T & \boldsymbol{0}\end{bmatrix}^T + \begin{bmatrix}\boldsymbol{N} & \boldsymbol{E}\\ -\boldsymbol{E}^T & \boldsymbol{0}\end{bmatrix}\right)\boldsymbol{y}\\ &= \boldsymbol{y}^{*T}\begin{bmatrix}2\boldsymbol{N} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}\boldsymbol{y} \geq 0\end{aligned} \tag{26}$$

is also nonnegative definite for any complex vector $\boldsymbol{y}$. From (24)–(26), it follows that the second passivity condition is satisfied. The reader should note that in this proof, no assumptions on the transformation matrix $\boldsymbol{X}$ had been made for none was needed. This gives the algorithm significant flexibility in choosing $\boldsymbol{X}$ to improve macromodel accuracy and/or run time.

B. Preservation of Moments

In this section, it will be shown that the transformation in (20) preserves $\lfloor q/N \rfloor$block moments of the original system, which is the same as the classical block Arnoldi reduction and half of that in MPVL. The block moments $\boldsymbol{M}_i$ of the original system, as shown in (7), are

$$\boldsymbol{M}_i = \boldsymbol{L}^T\boldsymbol{A}^i\boldsymbol{R} \tag{27}$$

where $\boldsymbol{A} \equiv -\boldsymbol{G}^{-1}\boldsymbol{C}$, $\boldsymbol{R} \equiv \boldsymbol{G}^{-1}\boldsymbol{B}$, and $\boldsymbol{G}$, $\boldsymbol{C}$, $\boldsymbol{B}$, $\boldsymbol{L}$ are the system matrices as defined in (1).

Likewise, the moments of the reduced-order system are given by

$$\hat{\boldsymbol{M}}_i = \tilde{\boldsymbol{L}}^T\tilde{\boldsymbol{A}}^i\tilde{\boldsymbol{R}} \tag{28}$$

where $\tilde{\boldsymbol{A}} \equiv -\tilde{\boldsymbol{G}}^{-1}\tilde{\boldsymbol{C}}$, $\tilde{\boldsymbol{R}} \equiv \tilde{\boldsymbol{G}}^{-1}\tilde{\boldsymbol{B}}$, and $\tilde{\boldsymbol{G}}$, $\tilde{\boldsymbol{C}}$, $\tilde{\boldsymbol{B}}$, $\tilde{\boldsymbol{L}}$ are as defined in (20). Substitution of (20) in (28) yields

$$\hat{\boldsymbol{M}}_i = \boldsymbol{L}^T\boldsymbol{X}\left[-\left(\boldsymbol{X}^T\boldsymbol{G}\boldsymbol{X}\right)^{-1}\left(\boldsymbol{X}^T\boldsymbol{C}\boldsymbol{X}\right)\right]^i\left(\boldsymbol{X}^T\boldsymbol{G}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{B}. \tag{29}$$

For a qth-order approximation, the columns of $\boldsymbol{X}$ span $Kr(\boldsymbol{A}, \boldsymbol{R}, q)$; therefore, it is shown in [6] that

$$\boldsymbol{A}^i\boldsymbol{R} = \boldsymbol{X}\boldsymbol{H}_q^i\boldsymbol{X}^T\boldsymbol{R}, \qquad 0 \leq i < \left\lfloor \frac{q}{N} \right\rfloor. \tag{30}$$

Rearranging the terms and using the definitions from (20)

$$\begin{aligned}\boldsymbol{A}\boldsymbol{A}^{i-1}\boldsymbol{R} &= \boldsymbol{X}\boldsymbol{H}_q^i\boldsymbol{X}^T\boldsymbol{R}\\ -\boldsymbol{G}^{-1}\boldsymbol{C}\boldsymbol{A}^{i-1}\boldsymbol{R} &= \boldsymbol{X}\boldsymbol{H}_q^i\boldsymbol{X}^T\boldsymbol{R}\\ -\boldsymbol{C}\boldsymbol{A}^{i-1}\boldsymbol{R} &= \boldsymbol{G}\boldsymbol{X}\boldsymbol{H}_q^i\boldsymbol{X}^T\boldsymbol{R}\\ -\boldsymbol{X}^T\boldsymbol{C}\boldsymbol{A}^{i-1}\boldsymbol{R} &= \boldsymbol{X}^T\boldsymbol{G}\boldsymbol{X}\boldsymbol{H}_q^i\boldsymbol{X}^T\boldsymbol{R}\end{aligned} \tag{31}$$

$$-\boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{G}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{C}\boldsymbol{A}^{i-1}\boldsymbol{R} = \boldsymbol{X}\boldsymbol{H}_q^i\boldsymbol{X}^T\boldsymbol{R}. \tag{32}$$

Inserting (30) in (32) results in

$$\boldsymbol{K}\boldsymbol{A}^{i-1}\boldsymbol{R} = \boldsymbol{A}^i\boldsymbol{R}, \qquad 0 \leq i < \left\lfloor \frac{q}{N} \right\rfloor \tag{33}$$

where

$$\boldsymbol{K} = -\boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{G}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{C}. \tag{34}$$

From (33), it can be shown by recursion that

$$\boldsymbol{K}^i\boldsymbol{R} = \boldsymbol{A}^i\boldsymbol{R}, \qquad 0 \leq i < \left\lfloor \frac{q}{N} \right\rfloor. \tag{35}$$

Therefore, using (34), it follows that

$$\boldsymbol{X}\left[-\left(\boldsymbol{X}^T\boldsymbol{G}\boldsymbol{X}\right)^{-1}\left(\boldsymbol{X}^T\boldsymbol{C}\boldsymbol{X}\right)\right]^i = \boldsymbol{K}^i\boldsymbol{X}. \tag{36}$$

Replacing $\boldsymbol{X}[-(\boldsymbol{X}^T\boldsymbol{G}\boldsymbol{X})^{-1}(\boldsymbol{X}^T\boldsymbol{C}\boldsymbol{X})]^i$ in (29) with $\boldsymbol{K}^i\boldsymbol{X}$ yields

$$\hat{\boldsymbol{M}}_i = \boldsymbol{L}^T\boldsymbol{K}^i\boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{G}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{B}. \tag{37}$$

Evaluating (30) when $i = 0$ yields

$$\boldsymbol{R} = \boldsymbol{X}\boldsymbol{X}^T\boldsymbol{R}. \tag{38}$$

Multiplying both sides by $\boldsymbol{X}^T\boldsymbol{G}$ gives

$$\boldsymbol{X}^T\boldsymbol{B} = \left(\boldsymbol{X}^T\boldsymbol{G}\boldsymbol{X}\right)\boldsymbol{X}^T\boldsymbol{R} \tag{39}$$

and it follows as

$$\boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{G}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{B} = \boldsymbol{R}. \tag{40}$$

Then, combining (37) and (40)

$$\hat{\boldsymbol{M}}_i = \boldsymbol{L}^T\boldsymbol{K}^i\boldsymbol{R}, \qquad 0 \leq i < \left\lfloor \frac{q}{N} \right\rfloor. \tag{41}$$

Finally, comparing (35) and (41) with (27), it follows that

$$\hat{\boldsymbol{M}}_i = \boldsymbol{M}_i, \qquad 0 \leq i < \left\lfloor \frac{q}{N} \right\rfloor. \tag{42}$$

Note that the number of poles in each entry of $\boldsymbol{Y}(s)$ is q, and we have matched the first $\lfloor q/N \rfloor$ moments at all N ports, yielding a total of q moments. In fact, it can be easily shown that if $\boldsymbol{X}$ spans the kth moment vector that is generated by the jth source, then the kth columns of the jth block moments of $\hat{\boldsymbol{Y}}(s)$ and $\boldsymbol{Y}(s)$ match as in block Arnoldi. The number of moments matched in this particular implementation of PRIMA is, therefore, the same as that for the block Arnoldi algorithm and half as many as matched by the block Lanczos algorithm.

C. Connection to Block Arnoldi

The block-Arnoldi-based PRIMA produces a reduced-order model that is quite similar to that of the block Arnoldi process, but slightly different in ways which seem to improve the accuracy. We can express the PRIMA reduced-order system as in these system equations:

$$\begin{aligned}-\tilde{\boldsymbol{G}}^{-1}\tilde{\boldsymbol{C}}\dot{\tilde{\boldsymbol{x}}}_q &= \tilde{\boldsymbol{x}}_q - \tilde{\boldsymbol{G}}^{-1}\tilde{\boldsymbol{B}}\boldsymbol{u}_N\\ \boldsymbol{i}_N &= \tilde{\boldsymbol{L}}^T\tilde{\boldsymbol{x}}_q.\end{aligned} \tag{43}$$

The poles of the reduced-order system in the block Arnoldi process are the inverses of the eigenvalues of $\boldsymbol{H}_q$ from (9). In PRIMA, however, the poles are the inverses of eigenvalues of $-\tilde{\boldsymbol{G}}^{-1}\tilde{\boldsymbol{C}}$. In this section, we show that the matrices $\boldsymbol{H}_q$ and $-\tilde{\boldsymbol{G}}^{-1}\tilde{\boldsymbol{C}}$ are identical, except for the last N columns (N is the number of ports) and the relation $\tilde{\boldsymbol{G}}^{-1}\tilde{\boldsymbol{B}} = \boldsymbol{X}^T\boldsymbol{R}$.

Multiplying both sides with $\boldsymbol{X}^T$ in (30) gives

$$\boldsymbol{X}^T\boldsymbol{A}^i\boldsymbol{R} = \boldsymbol{H}_q^i\boldsymbol{X}^T\boldsymbol{R}, \qquad 0 \le i < \left\lfloor \frac{q}{N} \right\rfloor. \tag{44}$$

Inserting (35) into (44) yields

$$\boldsymbol{X}^T\boldsymbol{K}^i\boldsymbol{R} = \boldsymbol{H}_q^i\boldsymbol{X}^T\boldsymbol{R}, \qquad 0 \le i < \left\lfloor \frac{q}{N} \right\rfloor. \tag{45}$$

Inserting (38) into (45) results in

$$\boldsymbol{X}^T\boldsymbol{K}^i\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{R} = \boldsymbol{H}_q^i\boldsymbol{X}^T\boldsymbol{R}, \qquad 0 \le i < \left\lfloor \frac{q}{N} \right\rfloor. \tag{46}$$

We know from (36) that

$$\boldsymbol{X}^T\boldsymbol{K}^i\boldsymbol{X} = \left[-\left(\boldsymbol{X}^T\boldsymbol{G}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{C}\boldsymbol{X}\right]^i = \left[\tilde{\boldsymbol{G}}^{-1}\tilde{\boldsymbol{C}}\right]^i. \tag{47}$$

Using (47) in (46) gives

$$\left[-\tilde{\boldsymbol{G}}^{-1}\tilde{\boldsymbol{C}}\right]^i\boldsymbol{X}^T\boldsymbol{R} = \boldsymbol{H}_q^i\boldsymbol{X}^T\boldsymbol{R}, \qquad 0 \le i < \left\lfloor \frac{q}{N} \right\rfloor. \tag{48}$$

Since $\boldsymbol{X}$ is obtained via a block Arnoldi process, $\boldsymbol{X}^T\boldsymbol{R}$ is only nonzero in the first N rows since the other rows of $\boldsymbol{X}^T$ are built to be orthonormal to $\boldsymbol{R}$:

$$\boldsymbol{X}^T\boldsymbol{R} = \begin{bmatrix}\boldsymbol{T}^T_{N\times N} & \boldsymbol{0}_{N\times N} & \boldsymbol{0}_{N\times N} & \cdots\end{bmatrix}^T \tag{49}$$

where $\boldsymbol{T}$ is the $N \times N$ upper triangular full rank matrix produced in the fourth step of the PRIMA algorithm given in Fig. 4. Therefore, left multiplication of any matrix by $\boldsymbol{X}^T\boldsymbol{R}$ only extracts the information about the first N columns of the matrix.

Hence, evaluating (48) when $i = 1$ means that the first N columns of $\boldsymbol{H}_q$ and $-\tilde{\boldsymbol{G}}^{-1}\tilde{\boldsymbol{C}}$ are the same. Since $\boldsymbol{H}_q$ is an upper block Hessenberg matrix (with blocks of $N \times N$), the first N columns of $\boldsymbol{H}_q^i$ (for $i > 1$) depend only on the first $(i-1)N$ columns of $\boldsymbol{H}_q$. From this information, evaluating (48) from $i = 1$ to $\lfloor q/N \rfloor - 1$ recursively demonstrates that the columns of $-\tilde{\boldsymbol{G}}^{-1}\tilde{\boldsymbol{C}}$ and $\boldsymbol{H}_q$ are the same until the last N columns.

To demonstrate that $\tilde{\boldsymbol{G}}^{-1}\tilde{\boldsymbol{B}} = \boldsymbol{X}^T\boldsymbol{R}$, we can simply multiply both sides of (40) by $\boldsymbol{X}^T$ and use the relation $\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{I}_q$.

IV. Time-Domain Simulation of the Macromodels

For a complete simulation, the nonlinear elements should be simulated along with the reduced-order macromodels. The popular and reliable simulation tool for general nonlinear circuits is SPICE [15]. Here, we describe two ways to include the PRIMA macromodels into circuit simulators such as SPICE. The first technique is direct stamping of the reduced-order matrices into the general SPICE MNA matrix, whereas the second method is based on a y-parameter description of the macromodel. The first method can also lead to an equivalent SPICE netlist production as a result of the reduction.

A. Direct Stamping and Realization

Since the reduced-order circuit is described in real matrices, it can be directly stamped into the SPICE MNA matrices. Noticing that the reduced-order q-variable system has the governing equations as in (43), and recognizing that it is possible to introduce $\tilde{\boldsymbol{x}}_q$ as a circuit variable into the MNA matrix, the direct stamps for the macromodel can be generated as

$$\begin{bmatrix} \text{Stamps for} & \boldsymbol{0} & \boldsymbol{0} \\ f(x_{NL}, u_p) & \boldsymbol{I}_N & \boldsymbol{0} \\ \boldsymbol{0} \quad \boldsymbol{0} & \boldsymbol{I}_N & -\tilde{\boldsymbol{L}}^T \\ \boldsymbol{0} \quad (\tilde{\boldsymbol{G}}^{-1}\tilde{\boldsymbol{B}}) & \boldsymbol{0} & \left(\boldsymbol{I} + \tilde{\boldsymbol{G}}^{-1}\tilde{\boldsymbol{C}}\frac{d}{dt}\right) \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{NL} \\ \boldsymbol{u}_p \\ \boldsymbol{i}_p \\ \tilde{\boldsymbol{x}}_q \end{bmatrix} = \begin{bmatrix} \boldsymbol{v}_{NL} \\ \boldsymbol{v}_p \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}. \tag{50}$$

In (50), $\boldsymbol{x}_{NL}$ denotes the other variables of the circuit (other node voltages and currents), $\boldsymbol{u}_p$ and $\boldsymbol{i}_p$ are port voltages and currents, respectively, and $\tilde{\boldsymbol{x}}_q$ denotes the extra variables that are introduced from the inclusion of the realized macromodel into the circuit. The realized macromodel will introduce only $q + N$ unknowns to the SPICE MNA matrix.

At this point, we understand it is possible to postprocess the system in (43) to achieve superior performance or an improved implementation. Without postprocessing, the $\tilde{\boldsymbol{G}}^{-1}\tilde{\boldsymbol{C}}$ matrix will be block upper Hessenberg, which degrades the sparsity of the SPICE MNA matrix, particularly when N is large.

B. Y-Parameter-Based Simulation

In order to compute the y parameters of the reduced-order system, the eigendecomposition steps [(14) and (15)] are used. After finding the poles and residues for $\boldsymbol{Y}(s)$, convolution is needed since the finite-difference methods employed in SPICE are for time-domain analysis and the macromodel is described by its y parameters in the frequency domain. Specifically, the currents at the ports would be computed by

$$i_p(t) = \sum_{j=1}^{N} \int_0^t y_{pj}(t-\tau)v_j(\tau)\,d\tau, \qquad 1 \le p \le N \tag{51}$$

which requires $O(T^2)$ complexity, where T is the number of time points during simulation. For this reason, recursive convolution [16] and time-domain y-parameter macromodels [17] were developed, where the complexity is linear with the number of time points. Details of these approaches are available in [17] and [16].

V. Results

In this section, our passive reduction algorithm is demonstrated and compared with other approaches. For the frequency-domain examples, the y parameters are compared with the reduced-order models from different reduction methods. Time-domain results are obtained using a modified version of SPICE3f4 [18] to perform recursive convolution. For all of the examples, the poles obtained via PRIMA were

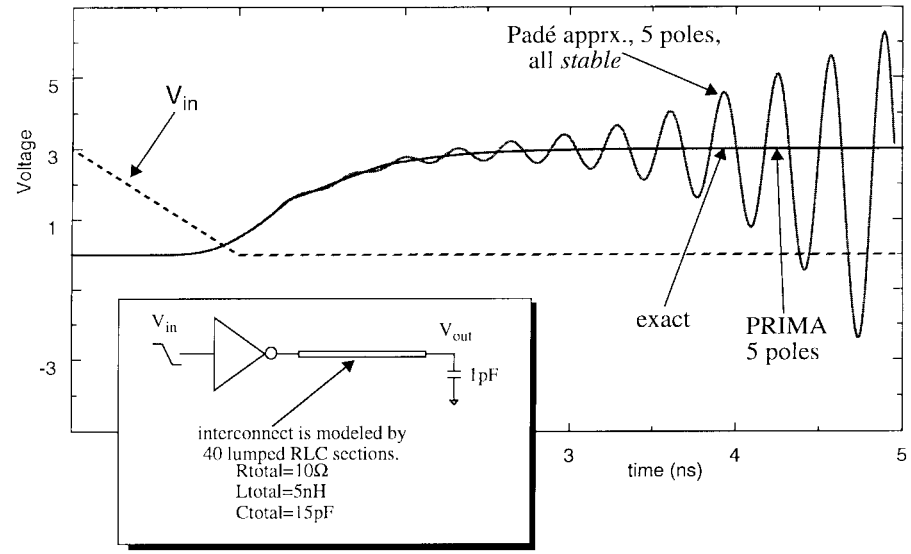

Fig. 5. Stable but nonpassive macromodel.

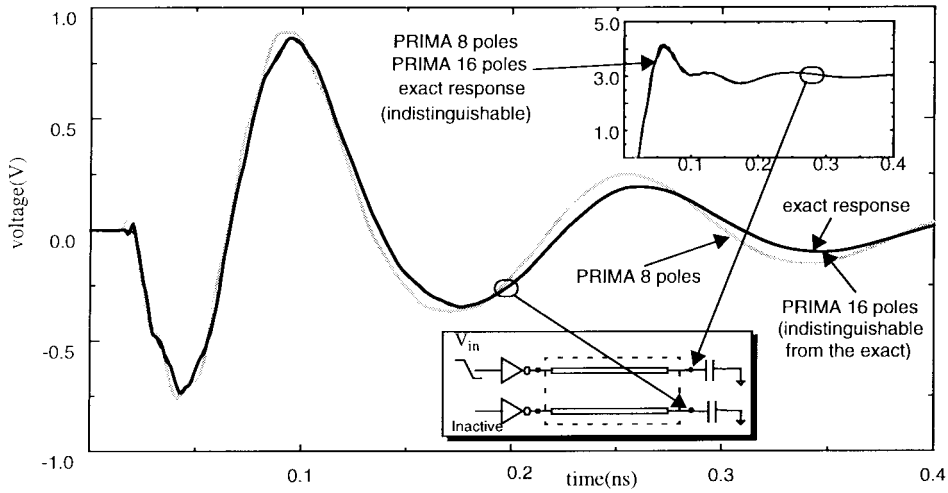

Fig. 6. Waveform comparisons for a four port.

observed to be *stable* at all times, which was a practical verification for the analyticity condition of $\hat{\boldsymbol{Y}}(s)$.

A. Nonlinear Driver Driving a Transmission Line

To demonstrate the importance of passivity, we considered the analysis of a lossy transmission line which was modeled by 40 lumped *RLC* sections. The model order reduction was performed by both PVL (Padé) and PRIMA using five poles. Although all of the poles from the Padé approximation were stable (i.e., negative real parts), the overall system was clearly unstable in Fig. 5. Note that the fifth-order approximation from PRIMA is indistinguishable from the exact response.

B. Coupled Noise For a 2-Bit Bus

In Fig. 6, a 2-bit bus driven by CMOS inverters is shown. One of the drivers is switching while the other is quiet. The interconnect, consisting of 40 coupled *RLC* sections, is modeled as a four-port and reduced by PRIMA. Transient

TABLE I
RUN TIME COMPARISONS OF DIRECT REALIZATION AND y-PARAMETER-BASED SIMULATIONS

Exact	Direct Realization	Y-parameter based
17.78 sec	0.6 with 8 poles	0.18 s. with 8 poles
	3.98 s. with 16 poles	0.28 s. with 16 poles
	10.29 s. with 24 poles	0.32 s. with 24 poles

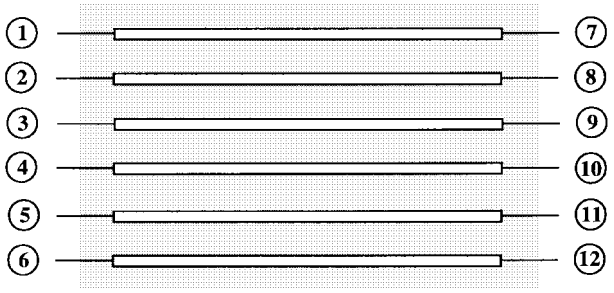

Fig. 7. Six coupled transmision lines forming a 12 port.

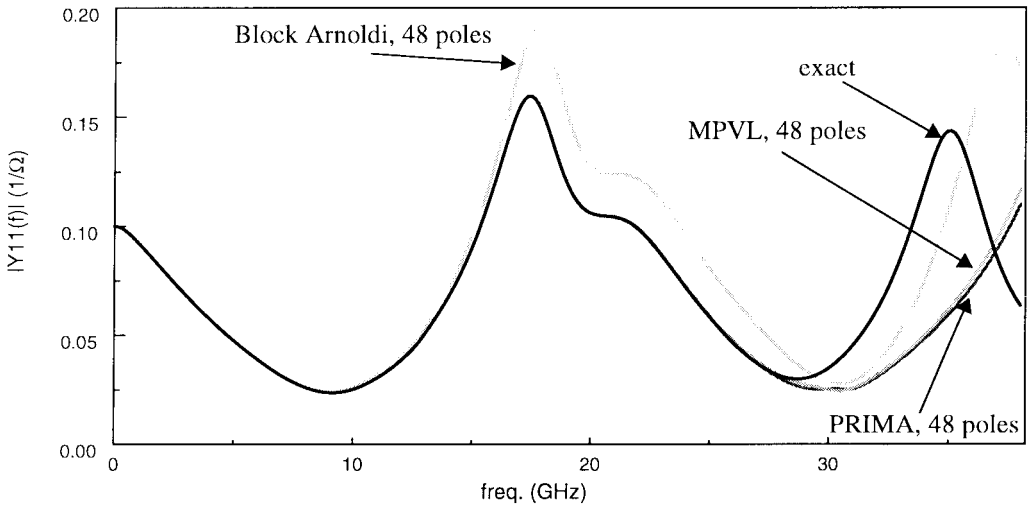

Fig. 8. $\boldsymbol{Y}_{11}(s)$ in frequency domain for six coupled TR lines.

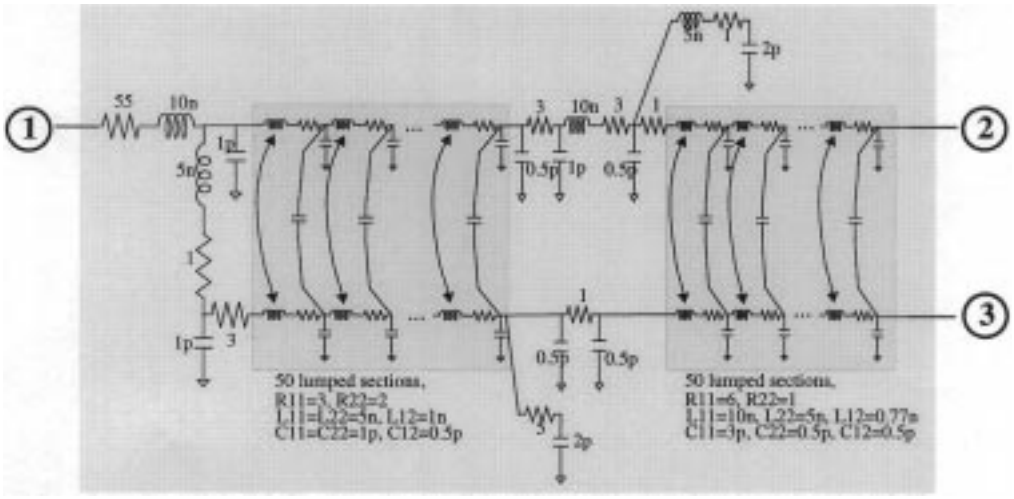

Fig. 9. Three-port consisting of a large lumped *RLC* circuit.

analysis is done using recursive convolution. The time-domain waveforms at the load end are compared for various order of approximations. Since this is a four-port, an eight-pole approximation corresponds to matching only m_0 and m_1 generated by four different sources. The plot shows that, in the time domain, even the coupled noise can be accurately simulated using the eight poles from PRIMA. Although, in this example, the inductance of the interconnect is exaggerated to make things worse, it is seen that an approximation of order 8 is enough to capture the coupled noise from the active driver to the quiet load end.

To compare the difference between direct realization and y-parameter-based simulation (i.e., recursive convolution here), the reduced-order circuit (via PRIMA) is simulated using both techniques. The run times are given in Table I. Although the circuit is relatively small (i.e., $\boldsymbol{G}$ is only 300 $\times$ 300), the gain in using PRIMA reduced macromodel and y-parameter-based simulation is about 50$\times$. For larger circuits, this gain is expected to be much larger.

C. Six Coupled Transmission Lines

As a second example, we analyzed a 12 port containing six coupled transmission lines modeled by 40 coupled *RLC* sections (Fig. 7). The input admittance $(Y_{11}(s))$, reduced by block Arnoldi, MPVL, and PRIMA, are compared with the exact input admittance original in Fig. 8 using 48 poles. Block Arnoldi captures the exact response up to 16 GHz, while MPVL and PRIMA match up to 28 GHz. When the order of approximation is increased to 72 poles, it is observed that the frequency spectrum is captured up to 60 GHz by MPVL and PRIMA.

D. Large Coupled RLC Circuit

The third example is a three-port, composed of a densely coupled *RLC* circuit shown in Fig. 9. Approximations were performed using 25 poles for the three methods. As can be observed from Fig. 10, PRIMA and MPVL captures the entire frequency spectrum.

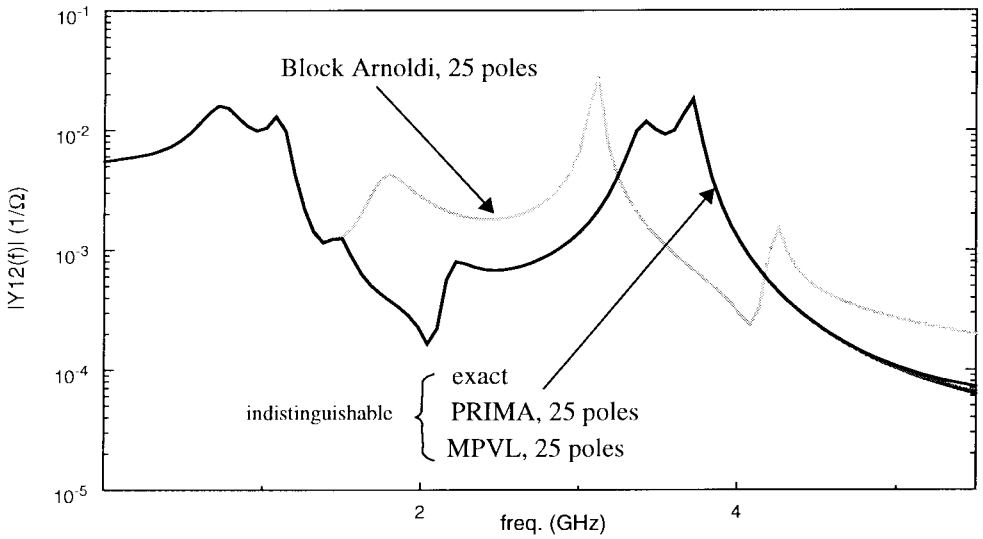

Fig. 10. $\boldsymbol{Y}_{12}(s)$ for the three-port in Fig. 9.

VI. Conclusion

This paper presented PRIMA, a novel algorithm for producing provably passive macromodels for arbitrary *RLC* circuits. A simple implementation of PRIMA given here uses the block Arnoldi algorithm to generate the vectors needed for applying the transformations to the MNA matrices. Results show that the approach tends to be comparable to MPVL in terms of frequency-domain accuracy, but superior in that it guarantees the passivity that is critical for time-domain analyzes. Using the same principles that were introduced in PRIMA, it is possible to obtain passive reduced-order models for general *RLC* circuits in several ways. Further extensions to PRIMA have already been demonstrated. In [13], the moment vectors from different frequency expansion points were used in forming the block Krylov space to increase the accuracy. In [8], it is demonstrated that it is possible to obtain the block Krylov space used in PRIMA via a *J*-symmetric Lanczos process with improvements in run time.

References

[1] 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.
[2] P. Feldmann and R. W. Freund, "Efficient linear circuit analysis by Padé approximation via the Lanczos process," *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 639–649, May 1995.
[3] K. J. Kerns, I. L. Wemple, and A. T. Yang, "Stable and efficient reduction of substrate model networks using congruence transforms," in *IEEE/ACM Proc. ICCAD*, Nov. 1995, pp. 207–214.
[4] L. M. Silveira, M. Kamon, and J. White, "Efficient reduced-order modeling of frequency-dependent coupling inductances associated with 3-D interconnect structures," in *IEEE/ACM Proc. DAC*, June 1995, pp. 376–380.
[5] P. Feldmann and R. W. Freund, "Reduced-order modeling of large linear subcircuits via a block Lanczos algorithm," in *IEEE/ACM Proc. DAC*, June 1995, pp. 474–479.
[6] D. L. Boley, "Krylov space methods on state-space control models," *Circuits Syst. Signal Processing*, vol. 13, no. 6, pp. 733–758, 1994.
[7] E. J. Grimme, "Krylov projection methods for model reduction," Ph.D. dissertation, Univ. Illinois, Urbana–Champaign, 1997.
[8] R. Freund, "Reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation," Numer. Anal. Manuscript 98-3-02, Bell Lab., Murray Hill, NJ, Feb. 1998.
[9] L. M. Silveira, M. Kamon, I. Elfadel, and J. White, "A coordinate-transformed Arnoldi algorithm for generating guaranteed stable reduced-order models of arbitrary RLC circuits," in *IEEE/ACM Proc. ICCAD*, Nov. 1996, pp. 288–294.
[10] K. J. Kerns, "Accurate and stable reduction of RLC networks using split congruence transformations," Ph.D. dissertation, Univ. Washington, Sept. 1996.
[11] B. D. Anderson and S. Vongpanitlerd, *Network Analysis and Synthesis*. Englewood Cliffs, NJ: Prentice-Hall, 1973.
[12] R. A. Rohrer and H. Nosrati, "Passivity considerations in stability studies of numerical integration algorithms," *IEEE Trans. Circuits Syst.*, vol. CAS-28, pp. 857–866, Sept. 1981.
[13] I. M. Elfadel and D. D. Ling, "A block rational Arnoldi algorithm for multipoint passive model-order reduction of multiport RLC networks," in *IEEE/ACM Proc. ICCAD*, Nov. 1997, pp. 66–71.
[14] E. S. Kuh and R. A. Rohrer, *Theory of Linear Active Networks*. San Francisco: Holden-Day, 1967.
[15] L. W. Nagel, "SPICE2, A computer program to simulate semiconductor circuits," Tech. Rep. ERL-M520, Univ. California, Berkeley, May 1975.
[16] V. Raghavan, J. E. Bracken, and R. A. Rohrer, "AWESpice: A general tool for the accurate and efficient simulation of interconnect problems," in *IEEE/ACM Proc. DAC*, June 1992, pp. 87–92.
[17] S. Y. Kim, N. Gopal, and L. T. Pillage, "Time-domain macromodels for VLSI interconnect analysis," *IEEE Trans. Computer-Aided Design*, vol. 13, pp. 1257–1270, Oct. 1994.
[18] T. L. Quarles, "The SPICE3 implementation guide," Tech. Rep. Memo ERL-M89/44, Univ. California, Berkeley, 1989.

Altan Odabasioglu received the B.S. degree in electrical engineering from Bilkent University, Ankara, Turkey, in 1995, and the M.S. degree in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, in 1996.

He was a summer intern at Harris Semiconductors, Melbourne, FL, in 1996 and at Strategic CAD Labs at Intel Corporation, Hillsboro, OR, in 1997. He is currently with Monterey Design Systems, Sunnyvale, CA, and working toward the Ph.D. degree at Carnegie Mellon University. His research interests include numerical aspects of circuit simulation and interconnect analysis.

Mustafa Celik (S'89–M'90) received the B.S. degree from Middle East Technical University, Ankara, Turkey, in 1988, and the M.S. and Ph.D. degrees from Bilkent University, Ankara, Turkey, in 1991 and 1994, respectively, all in electrical engineering.

After graduation, he held Research Associate positions, first in the Department of Electrical and Computer Engineering, University of Arizona, Tucson, and then in the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA. In the summer of 1997, he was with Motorola, Inc. At the end of 1997, he joined the Pacific Numerix Corporation, Scottsdale, AZ, as a Staff Development Scientist. His research interests include VLSI interconnect analysis and circuit simulation.

Lawrence T. Pileggi (M'85–SM'94), for a photograph and biography, see p. 49 of the January 1998 issue of this TRANSACTIONS.