Model Order Reduction of Fully Parameterized Systems by Recursive Least Square Optimization

Zheng Zhang, [‡]Ibrahim (Abe) M. Elfadel, and Luca Daniel
Research Lab of Electronics, Massachusetts Institute of Technology
[‡]Masdar Institute of Science and Technology, United Arab Emirates
Email: z_zhang@mit.edu, [‡]ielfadel@masdar.ac.ae, luca@mit.edu

***Abstract*—This paper presents a method for the model order reduction of fully parameterized linear dynamic systems. In a fully parameterized system, not only the state matrices, but also can the input/output matrices be parameterized. This algorithm is based on neither conventional moment-matching nor balanced-truncation ideas; instead, it uses "optimal (block) vectors" to construct the projection matrix such that the system errors in the whole parameter space could be minimized. This minimization problem is formulated as a recursive least square (RLS) optimization and then solved at a low cost. Our algorithm is tested by a set of multi-port multi-parameter cases with both intermediate and large parameter variations. The numerical results show that high accuracy is guaranteed, and that very compact models can be obtained for multi-parameter models due to the fact that the ROM size is independent of the parameter number in our flow.**

I. Introduction

Design and process parameter variations need very careful consideration in submicron digital, mixed-signal and RF analog integrated circuit design. Parameterized model order reduction (PMOR) techniques can generate compact models reflecting the impact induced by design or process variations. Such techniques have become highly desirable in the EDA community in order to accelerate the time-consuming design space exploration, sensitivity analysis and automatic synthesis.

Several PMOR techniques have been developed for both linear and nonlinear circuits. Most are based on moment matching techniques [1]–[11] due to the numerical efficiency. These algorithms normally assume that the closed forms of the parameterized state-space models are given, or that the parameters' statistical distributions are known. With similar assumptions, the positive-real balanced truncation method [12] has been modified for parameterized interconnect model reduction [13], [14]. In many cases the designers do not know the exact symbolic forms of the parameterized circuit equations. As a result, neither moment matching nor positive-real balanced truncation can be used in the PMOR flow. A numerically efficient and flexible method is the variational PMTBR scheme [15], which starts from the state-space model and uses a cheap sampling scheme to approximate the Gramian matrix. This approach is capable of preserving passivity for symmetric systems (such as *RC* circuits), but not for general *RLC* interconnect models. When the system equations are not available, one can treat the original model as a black box, and then use system identification techniques to construct macromodels from simulated or measured data by, for example, the quasi-convex optimization method [16], [17].

In all of the above-mentioned PMORs, the input and output matrices are non-parameterized. This assumption is true for many model order reduction (MOR) problems, when the input (or output) matrix only represents the fixed positions of the excitation (or output) signals. Nevertheless, if we attempt to analyze the higher-order nonlinearity of parameterized analog/RF circuits, the input matrices of the resulting linearized model may also be parameter-dependent. A typical example comes from the Volterra-series based nonlinear circuit modeling [18]–[22]. The input matrix can also be parameterized when the input signal is posed into a network after being processed by a parameter-dependent block (such as a MEMS sensor). Unfortunately, for these systems that involve both parameterized state matrices and parameterized input/output matrices, few MOR solutions have been reported in the EDA community.

This paper proposes a method for the MOR of fully parameterized systems with possibly parameter-dependent input/output matrices. This method is not based on existing moment matching or balanced truncation. Instead, it generates some "optimal (block) vectors" by recursive least square (RLS) optimization to construct the projection matrix, through minimizing the error in the whole parameter space (c.f. Section III). By this error minimization procedure, high accuracy can be guaranteed; very small reduced-order models (ROM) can be obtained for multi-parameter models since the ROM size does not depend on parameter numbers in our flow. When large parameter variations are involved, the RLS method can generate highly accurate ROMs after parameter space partitioning (c.f. Section IV). Our algorithm does not need the closed forms of the parameterized circuit equations, thus it has wider application areas than existing moment matching based PMORs. Besides, since Taylor expansion is not used to approximate the parameterized system matrices, the positive semi-definite structures of some systems are preserved, which implies passivity preservation for parameterized *RLC* interconnect models.

II. Motivation and Problem Formulation

A. Fully Parameterized Linear Dynamics

In this paper, we consider the multi-parameter linear time-invariant (LTI) parameterized dynamic system

$$C(\lambda)\dot{x}(t) = G(\lambda)x(t) + B(\lambda)u(t), \quad y(t) = L(\lambda)x(t) \tag{1}$$

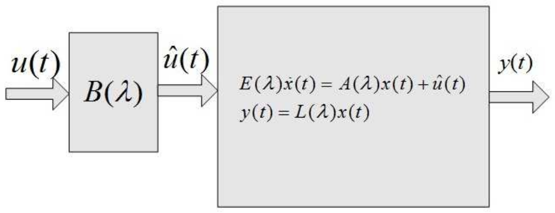


Fig. 1. Block diagram of the fully parameterized LTI system. The input $u(t)$ is converted by a parameter-dependent transformer before being fed to an LTI network which has n input ports. This system is a fully parameterized LTI model when we consider the signal path from $u(t)$ to $y(t)$.

where $C(\lambda), G(\lambda) \in \mathbb{R}^{n\times n}$ are state matrices, $B(\lambda) \in \mathbb{R}^{n\times m}$ and $L(\lambda) \in \mathbb{R}^{m\times n}$ are the input and output matrices, respectively. Here $\lambda \in \mathbb{R}^p$ is a vector representing p design or process parameters that influence the system equations.

The input and output matrices in (1) can also be parameterized, which were assumed non-parameterized in the previous literatures. To distinguish our model from the ones with non-parameterized input/output matrices, we refer to (1) as a **fully parameterized model**. The fully parameterized model can be interpreted as follows: the input signal $u(t)$ is first converted to $\hat{u}(t) \in \mathbb{R}^n$ by a transformer $B(\lambda)$, whose behavior depends on some parameters λ, before being injected into an LTI network, as shown in Fig. 1. The parameter-dependent transformer block $B(\lambda)$ can be many kinds of signal processing units in the real world, such as a voltage controlled current source, a MEMS sensor that converts chemical or optical signals to electrical signals, etc.

Fully parameterized models may also appear when we analyze some systems with non-parameterized input matrices. For example, in RF circuit analysis the parameter-dependent nonlinear circuit equation

$$\dot{q}(x(t),\lambda) = f(x(t),\lambda) + Bu(t) \tag{2}$$

can be written as the Volterra series

$$E(\lambda)\frac{dx}{dt} = A_1(\lambda)x + A_2(\lambda)x\otimes x + A_3(\lambda)x\otimes x\otimes x + Bu(t) \tag{3}$$

around the operating point, where $A_i(\lambda) \in \mathbb{R}^{n\times n^i}$ is the ith-order derivative matrix and $\otimes$ denotes the Kronecker tensor product. It has been shown in [18], [19] that the first- through third-order responses components of x can be obtained by solving the following linear systems, respectively,

$$\begin{aligned} E(\lambda)\dot{x}_1 &= A_1(\lambda)x_1 + Bu(t) \\ E(\lambda)\dot{x}_2 &= A_1(\lambda)x_2 + A_2(\lambda)x_1\otimes x_1 \\ E(\lambda)\dot{x}_3 &= A_1(\lambda)x_3 + A_2(\lambda)\left(x_1\otimes x_1 + x_2\otimes x_1\right) \\ &\quad + A_3(\lambda)x_1\otimes x_1 \end{aligned} \tag{4}$$

which are very useful for signal distortion or inter-modulation analysis in analog/RF circuit design. Obviously, the linear systems corresponding to x_2 and x_3 are fully parameterized LTI systems with parameter-dependent input matrices.

More interesting fully parameterized models can be found in the real-world applications.

B. Fully Parameterized Model Order Reduction

To speedup the computer-aided simulation, we aim at constructing a very small (for example, order-q, with $q \ll n$) fully parameterized reduced-order model (ROM)

$$C_r(\lambda)\dot{z}(t) = G_r(\lambda)z(t) + B_r(\lambda)u(t), \quad y_r(t) = L_r(\lambda)z(t). \tag{5}$$

The resulting ROM should approximate the original input/output relationship with high accuracy when the parameter λ varies in a specific range. A popular method is to select an appropriate projection matrix V to construct the ROM by

$$\begin{aligned} &C_r(\lambda) = V^TC(\lambda)V,\ G_r(\lambda) = V^TG(\lambda)V, \\ &B_r = V^TB(\lambda),\ L_r = L(\lambda)V, \end{aligned} \tag{6}$$

such that the ROM's parameterized transfer function

$$H_r(s,\lambda) = L_r(\lambda)(sC_r(\lambda) - G_r(\lambda))^{-1}B_r(\lambda) \tag{7}$$

is close to the original one for any λ within a specified variation range. Many PMOR methods have been proposed to reduce systems with non-parameterized input and output matrices, such as moment matching [1]–[4], [6], [7], [9], [11], positive-real balanced truncation [13], [14] and the sampling based variational PMTBR [15].

To the best knowledge of the authors, however, no results have been reported in the EDA community to reduce fully parameterized models whose input and output matrices are also parameterized. In the following sections we will present a fully parameterized MOR algorithm subject to the following requirements:

- It is feasible even when the input/output matrix is parameterized;
- It is applicable when the symbolic expression of the system matrices are unknown, which usually happens when (1) is extracted from transistor SPICE netlist and λ are some parameters in the complex semiconductor device model, or when the parameter dependence is shown by a look-up table or a measurement data sheet;
- Our technique is applicable even when the parameter variation is large, and it generates a ROM with size independent of the number of parameters;
- It preserves passivity for the positive semi-definite structured MNA (modified nodal analysis) equations.

III. Recursive Least Square based Fully PMOR

A. Main Idea

In our approach we assume that all entries of the parameterized system matrices are continuous functions of λ. This assumption is reasonable for many physical systems, and it is less conservative than the assumptions in previous PMOR algorithms where all entries in the system matrices are given as symbolic forms or every entry is differentiable with respect to (w.r.t) λ in the moment matching flows[1].

[1]Consider the simple parameterized matrix $E = E_0 + |\lambda|E_1$, and let λ be a scalar. This parameterized matrix is continuous at any $\lambda \in \mathbb{R}$. But we cannot use Taylor expansion around the nominal matrix E_0, because $|\lambda|$ is not differentiable at $\lambda = 0$.

In the frequency domain, we have

$$sC(\lambda)x(s,\lambda) = G(\lambda)x(s,\lambda) + B(\lambda)u(s). \tag{8}$$

We know that the transfer function from $u(s)$ to the state variable $x(s,\lambda)$, denoted as $X(s,\lambda)$, is obtained from solving the linear equation

$$(sC(\lambda) - G(\lambda))\, X(s,\lambda) = B(\lambda). \tag{9}$$

Inspired by the rational Krylov subspace method [23] for the linear MORs of non-parameterized systems, we know that if we can get $X(s,\lambda)$ at a set of frequency points and form V such that

$$\text{colspan}\left\{X(s_1,\lambda), X(s_2,\lambda), \cdots, X(s_p,\lambda)\right\} \subseteq V, \tag{10}$$

then an accurate ROM can be constructed. Unfortunately, we cannot compute the closed form of the **parameterized** (block) vectors $X(s_i,\lambda)$'s. Furthermore, it is normally impossible to orthonormalize these vectors even if they are computable for some specially-structured parameterized systems.

Alternatively, we can seek for a set of **non-parameterized** (block) vector $\hat{X}(s_i)$'s that are "close" to $X(s_i,\lambda)$'s under some accuracy criteria. If we substitute any $X(s_i) \in \mathbb{C}^{n\times m}$ back into (9), a parameter-dependent residual (block) vector is obtained:

$$e(s_i,\lambda) = (sC(\lambda) - G(\lambda))\, X(s_i) - B(\lambda), \tag{11}$$

which could be used for error estimation. Then we can seek for a non-parameterized (block) vector $\hat{X}(s_i)$ "close to" $X(s_i,\lambda)$, such that $\hat{X}(s_i)$ is an optimizer to the following problem

$$\begin{aligned}&\min \int_{\mathbb{S}} \left\|(s_iC(\lambda) - G(\lambda))\, X(s_i) - B(\lambda)\right\|_F^2 d\lambda(1)\cdots d\lambda(p),\\ &\text{subject to } X(s_i) \in \mathbb{C}^{n\times m}.\end{aligned} \tag{12}$$

Here $\mathbb{S} \subseteq \mathbb{R}^p$ denotes the parameter space, $\lambda(i)$ is the i-th coordinate of λ, and $\|M\|_F$ represents the Frobenius norm of a matrix $M \in \mathbb{C}^{n_1\times n_2}$:

$$\|M\|_F \triangleq \left(\sum_{j=1}^{n_2}\sum_{i=1}^{n_1} |M(i,j)|^2\right)^{\frac{1}{2}}. \tag{13}$$

The optimization problem (12) implies that $\hat{X}(s_i)$ is the non-parameterized (block) vector that is "nearest" to the parameterized $X(s_i,\lambda)$, if $\hat{X}(s_i)$ is selected such that the error [which is measured by the Frobenius norm of the residual matrix in (11)] is minimized in the whole parameter space.

Algorithm 1 Fully PMOR Flow

```
1: Solve the optimization problem (12) at frequency point $s_i$
   to obtain $\hat{X}(s_i)$ for $i = 1, 2, \cdots, q$
2: Construct the projection matrix $V$ such that
   range($V$)=colspan$\left\{\hat{X}(s_1), \hat{X}(s_2), \cdots, \hat{X}(s_q)\right\}$
3: Construct the fully parameterized ROM by (6).
```

Our fully PMOR is summarized in Algorithm 1. The key point is that the set of (block) vectors used to construct the projection matrix are obtained by solving an error minimization problem, instead of using the conventional moment matching, balanced truncation or sampling schemes. The main difficulty of this fully PMOR flow is the numerical solution of the optimization problem (12).

B. Reformulating Problem (12)

Since the parameterized system matrices are assumed to be continuous functions of λ, the integration in (12) can be computed after discretizing the parameter space. Note that the variation range of each element of λ can be characterized by its lower bound and upper bound, then the whole parameter space can be discretized by the following procedures.

1) The whole parameter space is represented by two vectors α and β of length p, such that $\mathbb{S} = \{\lambda | \alpha(i) \le \lambda(i) \le \beta(i), \text{ for i} = 1, \cdots, \text{p}\}$.
2) For each parameter $\lambda(i)$, its variation range (i.e., $[\alpha(i), \beta(i)]$)can be segmented into m_i uniform intervals. For $\lambda(i)$, the length of each interval is decided as
$$\Delta_i = \frac{\beta(i) - \alpha(i)}{m_i}. \tag{14}$$
3) Meanwhile, the above segmentation procedure would partition the whole parameter space $\mathbb{S}$ into $N = m_1 m_2 \cdots m_p$ boxes, each of which is p-dimension. For each integer $k \in [1, N]$, we can find a unique "companion vector" $\chi_k = [k_1, \cdots, k_p]$, based on the following rules:
   a) $k_1 + k_2 m_1 + k_3 m_2 m_1 + \cdots + k_p m_{p-1} m_{p-2} \cdots m_1 = k - 1$;
   b) $k_i \in \mathbb{Z}$ and $0 \le k_i \le m_i - 1$.

   Algorithm 2 has given the pseudo codes of calculating χ_k. With χ_k, the k-th p-dimensional box $\mathcal{B}(\chi_k)$ can be specified as
$$\alpha(i) + k_i\Delta_i \le \lambda(i) \le \alpha(i) + (k_i + 1)\Delta_i, \text{ for } i = 1, \cdots, p \tag{15}$$
   in the p-dimensional parameter space $\mathbb{S}$.
4) The volume of each box is
$$\Delta = \Delta_1\Delta_2\cdots\Delta_p. \tag{16}$$
5) The geometric center of the k-th box $\mathcal{B}(\chi_k)$ is a point $\bar{\lambda}(\chi_k)$ in the p-dimensional space, whose i-th coordinate is $\alpha(i) + (k_i + 0.5)\Delta_i$.

After discretization, the integration in problem (12) becomes

$$\begin{aligned}&\int_{\mathbb{S}} \left\|(s_iC(\lambda) - G(\lambda))\, X(s_i) - B(\lambda)\right\|_F^2 d\lambda(1)\cdots d\lambda(p)\\ &\approx \sum_{k=1}^{N} \Delta \left\|\left(s_iC\left(\bar{\lambda}(\chi_k)\right) - G\left(\bar{\lambda}(\chi_k)\right)\right) X(s_i) - B\left(\bar{\lambda}(\chi_k)\right)\right\|_F^2.\end{aligned} \tag{17}$$

Since Δ is a positive constant, now the original optimization can be reformulated as

$$\begin{aligned}&\min \sum_{k=1}^{N} \left\|\left(s_iC\left(\bar{\lambda}(\chi_k)\right) - G\left(\bar{\lambda}(\chi_k)\right)\right) X(s_i) - B\left(\bar{\lambda}(\chi_k)\right)\right\|_F^2\\ &\text{subject to } X(s_i) \in \mathbb{C}^{n\times m}.\end{aligned} \tag{18}$$

Problem (18) is solvable and can be easily tackled by the recursive least square (RLS) optimization in Section III-C.

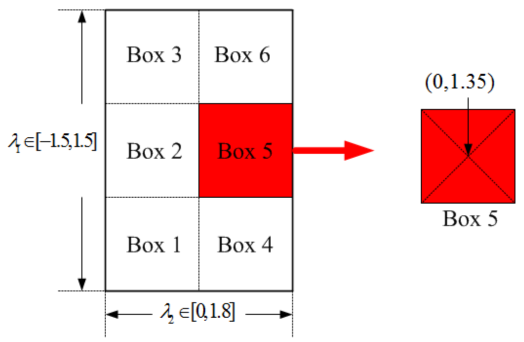


Fig. 2. Discretization example for a parameterized system with 2-D parameter space. Since $\lambda(1) \in [-1.5, 1.5]$ and $\lambda(2) \in [0, 1.8]$, we have $\alpha = [-1.5; 0]$ and $\beta = [1.5; 1.8]$. We use $m_1 = 3$ and $m_2 = 2$, which segments the range of $\lambda(1)$ into 3 intervals with $\Delta_1 = 1$ and the range of $\lambda(2)$ into 2 intervals with $\Delta_2 = 0.9$. Meanwhile, the whole parameter space $\mathbb{S}$ is partitioned into $3 \times 2 = 6$ two-dimension boxes (which are panels in 2-D parameter space), and each box's volume is $\Delta = 0.9$. For the 5th box, using Algorithm 2 we have $k-1 = 4 = 1 + 1 \times m_1$, therefore, the companion vector is $\chi_5 = [1, 1]$. With χ_5, the center of the 5th box is fixed as $\lambda(1) = \alpha(1) + 1.5\Delta_1 = 0$ and $\lambda(2) = \alpha(2) + 1.5\Delta_2 = 1.35$.

To illustrate the discretization flow, an example with a 2-D parameter space is given in Fig. 2. The extension to 3-D and higher-dimension parameter spaces is straightforward.

Algorithm 2 Compute the companion vector χ_k of k.

```
1: Initialize $\vec{m} \in \mathbb{Z}^p$: $\vec{m}(1) = 1$, $\vec{m}(i+1) = m_i \vec{m}(i)$ for $i = 1, \cdots, p-1$;
2: Initialize $\chi_k \in \mathbb{Z}^p$: $\chi_k = [0, \cdots, 0]$ and set $\hat{k} = k - 1$;
3: for $\hat{p} = p, \ldots p-1, \cdots, 1$ do
       $\chi_k(\hat{p}) = \mathrm{floor}(\frac{\hat{k}}{\vec{m}(\hat{p})})$, $\hat{k} = \hat{k} - \chi_k(\hat{p})\vec{m}(\hat{p})$.
```

C. RLS Optimization

In this section we present a method to solve problem (18).

First, we denote the j-th columns of $X(s_i)$ and $B\left(\bar{\lambda}(\chi_k)\right)$ by $X_j(s_i)$ and $B_j\left(\bar{\lambda}(\chi_k)\right)$, respectively. It is straightforward to show that

$$\begin{aligned} &\left\|\left(s_i C\left(\bar{\lambda}(\chi_k)\right) - G\left(\bar{\lambda}(\chi_k)\right)\right) X(s_i) - B\left(\bar{\lambda}(\chi_k)\right)\right\|_F^2 \\ &= \sum_{j=1}^{m} \left\|\left(s_i C\left(\bar{\lambda}(\chi_k)\right) - G\left(\bar{\lambda}(\chi_k)\right)\right) X_j(s_i) - B_j\left(\bar{\lambda}(\chi_k)\right)\right\|_2^2, \end{aligned} \tag{19}$$

where $\|b\|_2 = \left(\sum_{i=1}^{n} |b(i)|^2\right)^{\frac{1}{2}}$ is the 2-norm for any vector $b \in \mathbb{C}^n$. As a result, problem (18) can be rewritten as

$$\begin{aligned} &\min \sum_{j=1}^{m} \left(\sum_{k=1}^{N} \left\|\left(s_i C\left(\bar{\lambda}(\chi_k)\right) - G\left(\bar{\lambda}(\chi_k)\right)\right) X_j(s_i) - B_j\left(\bar{\lambda}(\chi_k)\right)\right\|_2^2\right), \\ &\text{subject to } X_j(s_i) \in \mathbb{C}^n, \text{ for } j = 1, \cdots, m. \end{aligned} \tag{20}$$

Let

$$f_j\left(X_j(s_i)\right) = \sum_{k=1}^{N} \left\|\left(s_i C\left(\bar{\lambda}(\chi_k)\right) - G\left(\bar{\lambda}(\chi_k)\right)\right) X_j(s_i) - B_j\left(\bar{\lambda}(\chi_k)\right)\right\|_2^2 \tag{21}$$

It is clear that for any integer $r \neq j$ $(1 \leq r \leq m)$, the value of $f_j\left(X_j(s_i)\right)$ is independent of $X_r(s_i)$. Therefore, the cost function of (18) is minimized if $f_j\left(X_j(s_i)\right)$ is minimized for $j = 1, \cdots m$, with $X_j(s_i)$ as the variable. From this observation, it is clear that the j-th column of $\hat{X}(s_i)$ can be decided by finding the optimizer to the problem

$$\begin{aligned} &\min \sum_{k=1}^{N} \left\|\left(s_i C\left(\bar{\lambda}(\chi_k)\right) - G\left(\bar{\lambda}(\chi_k)\right)\right) X_j(s_i) - B_j\left(\bar{\lambda}(\chi_k)\right)\right\|_2^2 . \\ &\text{subject to } X_j(s_i) \in \mathbb{C}^n \end{aligned} \tag{22}$$

If problem (22) is solved for $j = 1, \cdots, m$, then all columns of the "optimal block vector" $\hat{X}(s_i)$ can be obtained.

Next we give the RLS theorem.

Theorem: *Let $e_i = A_i x - b_i$ for $i = 1, \cdots, k$, with $e_i, b_i \in \mathbb{C}^{n_1}$, $x \in \mathbb{C}^{n_2}$, $A_i \in \mathbb{C}^{n_1 \times n_2}$, $n_1 \geq n_2$ and $A_i^H A_i$ is invertible*[2]*. Then the optimal solution to*

$$\min \sum_{i=1}^{k} \|e_i\|_2^2, \quad \text{subject to } x \in \mathbb{C}^{n_2} \tag{23}$$

can be written as

$$\hat{x} = \left(\sum_{i=1}^{k} A_i^H A_i\right)^{-1} \left(\sum_{i=1}^{k} A_i^H b_i\right). \tag{24}$$

Proof: Denote $e = [e_i; e_2; \cdots; e_k]$, $b = [b_1; , b_2; \cdots; b_k]$ and $A = [A_i; A_2; \cdots; A_k]$, then (23) can be converted to the least-square problem

$$\min \|e\|_2^2, \quad \text{subject to } e = Ax - b, \; x \in \mathbb{C}^{n_2} \tag{25}$$

from which we have $\hat{x} = \left(A^H A\right)^{-1} \left(A^H b\right) = \left(\sum_{i=1}^{k} A_i^H A_i\right)^{-1} \left(\sum_{i=1}^{k} A_i^H b_i\right)$. ■

According to the above theorem, we know that the optimizer for the RLS problem in (22) is

$$\begin{aligned} \hat{X}_j(s_i) = &\left(\sum_{k=1}^{N} M_i\left(\bar{\lambda}(\chi_k)\right)^H M_i\left(\bar{\lambda}(\chi_k)\right)\right)^{-1} \\ &\times \sum_{k=1}^{N} M_i\left(\bar{\lambda}(\chi_k)\right)^H B_j\left(\bar{\lambda}(\chi_k)\right) \end{aligned} \tag{26}$$

where

$$M_i\left(\bar{\lambda}(\chi_k)\right) = s_i C\left(\bar{\lambda}(\chi_k)\right) - G\left(\bar{\lambda}(\chi_k)\right) \tag{27}$$

is a nonsingular matrix, if the original dynamical system is stable and $s \notin \mathbb{C}^-$. The nonsingularity of $M_i\left(\bar{\lambda}(\chi_k)\right)$ implies that $M_i\left(\bar{\lambda}(\chi_k)\right)^H M_i\left(\bar{\lambda}(\chi_k)\right)$ is positive definite, so the matrix inversion in (26) is well posed.

Finally, the "optimal (block) vector" $\hat{X}(s_i)$ is obtained as

$$\begin{aligned} \hat{X}(s_i) = &\left(\sum_{k=1}^{N} M_i\left(\bar{\lambda}(\chi_k)\right)^H M_i\left(\bar{\lambda}(\chi_k)\right)\right)^{-1} \\ &\times \sum_{k=1}^{N} M_i\left(\bar{\lambda}(\chi_k)\right)^H B\left(\bar{\lambda}(\chi_k)\right). \end{aligned} \tag{28}$$

D. Algorithm Summary

Assume that q frequency points $s_1, \cdots, s_q$ are used in the fully PMOR flow, the original system has p parameters, its model size is n, and for parameter i its variation range is

Algorithm 3 RLS-based Fully PMOR.

```
1:  Initialize $\mathcal{X} \leftarrow [\,]$;
2:  for $i = 1, \cdots, q$ do
3:      $\mathcal{M} = 0$, $\mathcal{B} = 0$;
4:      for $k = 1, \cdots, N$ do
5:          calculate the companion vector $\chi(k)$ of $k$;
6:          fix $\bar{\lambda}(\chi_k)$;
7:          $\mathcal{M} = \mathcal{M} + M_i\left(\bar{\lambda}(\chi_k)\right)^H M_i\left(\bar{\lambda}(\chi_k)\right)$;
8:          $\mathcal{B} = \mathcal{B} + M_i\left(\bar{\lambda}(\chi_k)\right)^H B\left(\bar{\lambda}(\chi_k)\right)$;
9:      end for
10:     compute $\hat{X}(s_i) = \mathcal{M}^{-1}\mathcal{B}$;
11:     $\mathcal{X} \leftarrow [\mathcal{X},\ \hat{X}(s_i)]$;
12: end for
13: orthonormalize the column vectors of $\mathcal{X}$ such that
    $\mathrm{range(V)} = \mathrm{colspan}\{\mathcal{X}\}$;
14: construct the fully parameterized ROM by (6).
```

uniformly segmented into m_i intervals. The details of our fully PMOR is are given in Algorithm 3.

The main computational cost comes from solving the linear system with m right-hand sides in Line 10, at a cost similar to that of Krylov-subspace based moment matching for non-parameterized systems [24]. The matrix-matrix and matrix-vector products in lines 7&8 are very cheap in circuit simulation, because the resulting matrices are normally very sparse. The proposed fully PMOR has the following properties:

- At each frequency point only one (block) vector $\hat{X}(s_i)$ is selected in the whole parameter space, so much redundant information is eliminated, which helps keep the ROM size small. Besides, the ROM size mq is also independent of the parameter number and the discretization scheme[3];
- We only need the numerical values of the system matrices when λ is the center of each box in the discretization scheme. Therefore, our algorithm is also applicable when the parameter dependence is given as a look-up table. For the models from semiconductor circuits, one can use SPICE to get the numerical values at the parameter points of interest, even if the parameter dependence inside the device model is not explicitly known;
- No Taylor expansion around the nominal value of λ is used, so the positive semi-definite structure of MNA equations can be preserved for interconnect simulation. This means that passivity can also be guaranteed.

IV. Dealing with Large Variations

A. Parameter Space Segmentation

When the parameter variation range is very large, the resulting ROM may not be as accurate as required. One solution is to use a finer discretization when computing $\hat{X}(s_i)$, by increasing the segmentation number of each parameter. This is helpful when the parameter space is "intermediate" or not "very" large. But when the discretization scheme is accurate enough, we cannot further improve the accuracy by using a finer discretization scheme. This is because the theoretically minimal error in (12) will grow when the parameter space is too large. In such a case, we can segment the whole parameter space into several smaller spaces before using Algorithm 3.

There does not exist a theoretically sound guideline on how to segment the original parameter space $\mathbb{S}$. One possible parameter space segmentation procedure is given below.

1) We first use Algorithm 3 to generate a fully parameterized ROM for the parameter space $\mathbb{S}$;
2) We see the center of $\mathbb{S}$ (denoted as $\hat{\lambda}_0 = \frac{\alpha+\beta}{2}$) as the nominal point. After that, we compute the transfer function of system (1) at a set of frequency points $s_1, \cdots, s_l$ in the specified frequency band, with $\lambda = \hat{\lambda}_0$. The computed "nominal" transfer functions are denoted as $H(s_j, \hat{\lambda}_0)$, with $j = 1, \cdots, l$.
3) We evaluate the effect induced by the variation of each parameter. For $\lambda(i)$, we increase λ's i-th coordinate from $\alpha(i)$ to $\beta(i)$ with a step size $\frac{\beta(i)-\alpha(i)}{r_i}$ (r_i is a positive integer), while keeping other coordinates nominal (i.e., identical to those of $\hat{\lambda}_0$). By doing so, we get r_i parameter points along the direction $\vec{e}_i$ in the parameter space ($\vec{e}_i$ is the i-th column of the $p \times p$ identity matrix). For each parameter point denoted by $\lambda_{i,\hat{k}}$ (with $\hat{k} = 1 \cdots, r_i$), we compute the ROM's transfer functions at the frequency points used in Step 2), and denote the result as $H_r(s_j, \lambda_{i,\hat{k}})$. Then we can get a $r_i \times l$ error matrix for parameter $\lambda(i)$
$$\Theta_i = \begin{bmatrix} \theta_{1,1} & \theta_{1,2} & \cdots & \theta_{1,l} \\ \theta_{2,1} & \theta_{2,2} & \cdots & \theta_{2,l} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{r_i,1} & \theta_{r_i,2} & \cdots & \theta_{r_i,l} \end{bmatrix}, \tag{29}$$
$$\text{with } \theta_{\hat{k},j} = \left\| H(s_j, \hat{\lambda}_0) - H_r(s_j, \lambda_{i,\hat{k}}) \right\|_F .$$
4) From Θ_i, we estimate the ROM's error that is induced by $\lambda(i)$'s variation by
$$\hat{\theta}_i = \max\ \theta_{\hat{k},j},\ \text{for } \hat{k} = 1, \cdots, r_i \text{ and } j = 1, \cdots, l \tag{30}$$
which represents the largest sampling error in the frequency-parameter space. We can also use another error estimator
$$\hat{\theta}_i = \frac{\left\| H(s_j, \hat{\lambda}_0) - H_r(s_j, \lambda_{i,\hat{k}}) \right\|_F^2}{r_i l} \tag{31}$$
which represents the average sampling error in the frequency-parameter space of $p+1$ dimensions.
5) We repeat the steps from 3) to 4) to get the error estimator for each parameter coordinate [i.e., $\hat{\theta}_1$ for $\lambda(1)$, $\cdots$, $\hat{\theta}_p$ for $\lambda(p)$].
6) We assume that $\hat{\theta}_k$ is the largest one among the obtained p error estimators. If $\hat{\theta}_k$ is larger than an error tolerance ϵ, we segment the range of $\lambda(i)$ into two sub-ranges $[\alpha(i), \frac{\alpha(i)+\beta(i)}{2}]$ and $[\frac{\alpha(i)+\beta(i)}{2}, \beta(i)]$. Meanwhile, the original parameter space $\mathbb{S}$ is portioned into two parts

[2] A_i^H denotes the conjugate transpose of A_i.

[3] For each fixed s_i, the sampling-based PMOR selects many (block) vectors to construct V, which introduces redundancy. In moment-matching PMORs, the ROM size grows exponentially with the parameter number p.

$\mathbb{S}_1$ and $\mathbb{S}_2$ by this segmentation. If $\hat{\theta}_k \leq \epsilon$, we do not partition the parameter space.

7) We repeat the above procedures to further partition the obtained smaller parameter spaces, if necessary.

The parameter space segmentation can also be performed with some empirical experience, which is less mathematically sound but normally effective. In practical problems, many parameters fluctuate in a small range (such as the dielectric constant of the metal interconnects, the threshold voltage of MOS transistors), and a few of others have larger variation ranges (e.g., the width and length sizing of interconnects or transistors in the design phase). Therefore, one can first segment the variation ranges of several parameters whose ranges are “obviously” large based on the design experience. If the result is not good enough, one can further consider the above-mentioned segmentation scheme.

B. Constructing Reduced-Order Models

After parameter space segmentations, we may get h sub parameter spaces $\mathbb{S}_1$, $\mathbb{S}_2$, $\cdots$, $\mathbb{S}_h$. For each of them one can use Algorithm 3 (precisely, Lines 1-12) to get h matrices $\mathcal{X}_k \in \mathbb{C}^{n\times mq}$, with $k = 1, 2, \cdots, h$. With these matrices, we can construct parameterized ROMs in different ways.

1) Multiple ROMs: each $\mathcal{X}_k$ is orthonormalized to form V_k, which is stored in a look-up table. Then using different V_k's, one can get h different ROMs by (6). We denote these ROMs by Σ_1, Σ_1, $\cdots$, Σ_h respectively. Then in the subsequent simulation, one can use Σ_k to perform fast ROM-based simulation if the the given parameter is inside the smaller parameter space $\mathbb{S}_k$. The parameter point λ may belong to several sub parameter spaces if it is located on the border. In such a case any Σ_k satisfying $\lambda \in \mathbb{S}_k$ can be used.

2) Single ROM: we first construct $\mathcal{X} = [\mathcal{X}_1, \mathcal{X}_2, \cdots, \mathcal{X}_h]$. To eliminate the possible redundant information, the dominant eigenspace of $\mathcal{X}\mathcal{X}^H$ is extracted to form the projection matrix V. Power iteration could be used to extract the dominant eigenspace of $\mathcal{X}\mathcal{X}^H$ [25], but it may be inaccurate when the largest eigenvalue of $\mathcal{X}\mathcal{X}^H$ is not distinct. Assuming that we want V to span the column space of the first $\hat{q}$ dominant eigenvectors of $\mathcal{X}\mathcal{X}^H$ ($\hat{q} < mhq \ll n$), the following procedures can be used:

- Perform singular value decomposition (SVD) for the much smaller-size matrix $\mathcal{X}^H\mathcal{X}$, obtaining
$$\mathcal{X}^H\mathcal{X} = U \begin{bmatrix} \delta_1 & & & \\ & \delta_2 & & \\ & & \ddots & \\ & & & \delta_{mhq} \end{bmatrix} U^H, \qquad (32)$$
since $\mathcal{X}^H\mathcal{X}$ is a Hermitian positive semi-definite matrix. In the SVD result, U is a unitary matrix, and $\delta_1 \geq \cdots \geq \delta_{mlq}$ are the singular values of $\mathcal{X}^H\mathcal{X}$.
- Compute the projection matrix V by
$$V = \mathcal{X}U_{\hat{q}} \begin{bmatrix} \frac{1}{\sqrt{\delta_1}} & & & \\ & \frac{1}{\sqrt{\delta_2}} & & \\ & & \ddots & \\ & & & \frac{1}{\sqrt{\delta_{\hat{q}}}} \end{bmatrix}, \qquad (33)$$
where $U_{\hat{q}}$ denotes the first $\hat{q}$ columns of U.

Through the SVD of $\mathcal{X}$, one can prove that $V^TV = I$ and $\mathrm{range(V)}$ spans the column space of the first q dominant eigenvectors of $\mathcal{X}\mathcal{X}^H$. When $mhq \ll n$, the cost of computing V is negligible. This procedure can also be used in Algorithm 3 to form V, if we need even smaller ROMs.

We use the second approach to generate single parameterized ROM. The pseudo codes are given as Algorithm 4.

Algorithm 4 RLS-based fully PMOR with Large Variations

```
1: Segment the original parameter space into $\mathbb{S}_1, \mathbb{S}_2, \cdots, \mathbb{S}_h$,
   by the procedures in Section IV-A;
2: Initialize $\mathcal{X} \leftarrow [\,]$;
3: for $i = 1, \cdots, h$ do
4:     Within the parameter space $\mathbb{S}_i$, use Lines 1-12 of
       Algorithm 3 to generate $\mathcal{X}_i$;
5:     $\mathcal{X} \leftarrow [\mathcal{X}, \mathcal{X}_i]$;
6: end for
7: Form $V$ by (32) and (33);
8: construct the fully parameterized ROM by (6).
```

V. Numerical Results

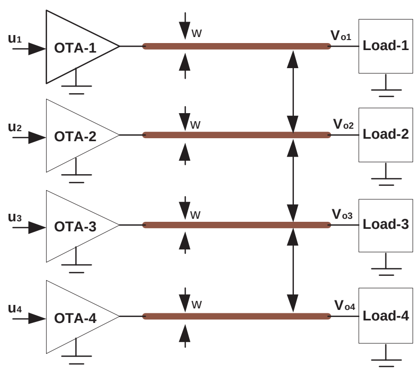
We use several multi-port fully parameterized examples with very large parameter variations to test the proposed RLS optimization-based PMOR scheme. All experiments are implemented in Matlab and performed on a 3.3GHz 4-GB RAM workstation.

A. Experimental Setup

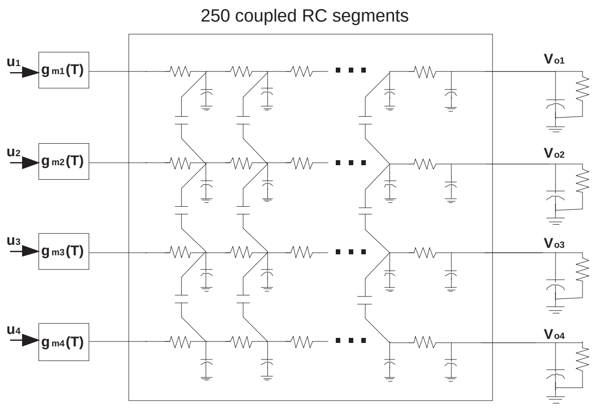
The circuit configuration in Fig 3 (a) is used to build multi-port multi-parameter fully parameterized benchmarks. OTA 1-4 are four operational transconductance amplifiers that convert input voltage signals to output currents. We assume that their f_{3dB} frequencies are much higher than the frequency we are interested in, then the OTA gains are some functions of the temperature T which influences transistor threshold voltages. Their gains are given as $g_{m1}(T)$, $g_{m2}(T)$, $g_{m3}(T)$ and $g_{m4}(T)$, respectively, by some data sheets. Four coupling metal lines connect the OTAs to the loading networks. We assume that the 4 lines have the same widths w, and the same spacing l between adjacent lines. As shown in Fig. 3 (b), 250 parameter-dependent coupled *RC* segments are used to model the metal lines, whose per-unit-length (p.u.l.) resistor, p.u.l. ground capacitor and p.u.l. coupling capacitor vary with w and l. Since the metal resistivity varies with temperature, the p.u.l. resistor also depends on T. Each loading network is modeled by a non-parameterized *RC* pair. For simplicity, we assume that the temperature profile is uniform. The input voltages are injected into the OTAs. We attempt to find the voltage gains from the OTA inputs to the far end of each metal line. The state-space equation can be written as the following fully parameterized model with four ports

$$C(w,l)\dot{x} = G(w,T)x + B(T)u, \;\; y = Lx.$$

Here $u = [u_1; u_2; u_3; u_4]$ and $y = [v_{o1}; v_{o2}; v_{o3}; v_{o4}]$ are shown in Fig. 3. The original problem size is 1004.



(a) Original circuit.



(b) Fully parameterized model.

Fig. 3. The multi-port fully parameterized example used to test our algorithm. (a) The original circuit consists of four ideal operational transconductance amplifiers (OTA), 4 coupling metal lines each loaded with a network. The excitation signals are injected into the OTAs, and we attempt to find the voltage gains from the inputs to the far ends of the interconnects. (b) The fully parameterized model used for simulation. The OTAs' gains depend on the temperature T. The interconnect parasitic *RC* performances are dependent on metal width (w), wire spacing (l) and the temperature T. The loading effect of each loading network is described by a non-parameterized *RC* pair.

B. Accuracy Verification of RLS-based Fully PMOR

We first test the RLS-based fully PMOR (Algorithm 3) with variations of moderate size. The parameter space $\mathbb{S}$ is listed in the 2nd row of Table I, where w_0, l_0 and T_0 denote the nominal parameter values. Three test cases are given in Rows 3-5.

TABLE I
PARAMETER SPACE SPECIFICATION AND TEST EXAMPLES FOR ALGORITHM 3.

	w/w_0	l/l_0	$T-T_0$
parameter space $\mathbb{S}$	$[1,30]$	$[1,15]$	$[0,100]$
case 1	3	1.5	10
case 2	15	7.5	50
case 3	27	13.5	90

In the RLS flow, 10 frequency points from 200Hz to 10^9 Hz are used. In the discretization scheme, each parameter range is divided into 2 intervals, which means that 8 "boxes" are used to compute each $\hat{X}(s_i)$. To make the ROM compact, we extract the dominant eigenspace of $\mathcal{X}\mathcal{X}^H$, and an order-10 parameterized ROM is generated for the parameter ranges specified in the first row of Table I. Fig. 4 gives the MOR results for test cases 1-3, where the ROMs' transfer functions are all indistinguishable from their original ones. When $f < 10^8$Hz, the relative errors are all below 10^{-8}.

For comparison, we also implemented the sampling based scheme in [15] with some modifications. In [15], the input matrix is fixed. In our experiments, the input matrix is updated as λ changes, but we do not change any other procedure of [15]. For each frequency point, 27 block vectors $X(s_i, \lambda)$ are produced at 27 parameter points. Three order-20 reduced models are generated for cases 1-3. The results in Fig. 5 (b) show that the relative errors from [15] are about 3 orders of magnitude higher than those from RLS optimization [c.f. Fig. 4 (b)], even though the ROM sizes from [15] is 2× larger than the results from our RLS-based fully PMOR.

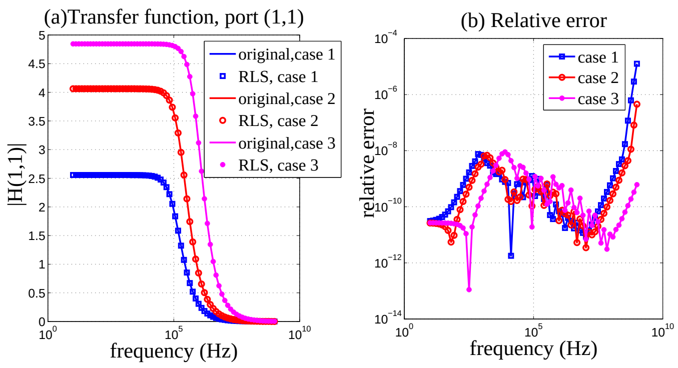


Fig. 4. Fully PMOR result by our RLS-based scheme (Algorithm 3) for cases 1-3, with ROM size=10. (a) The transfer functions (from u_1 to v_{o1}) of the obtained ROMs are indistinguishable from the original curves. (b) The maximum relative errors of cases 1-3 are about 10^{-5}, 10^{-6} and 10^{-8}, respectively.

C. Large-Variation Models

In this section we test Algorithm 4 on some models experiencing larger parameter variations (c.f. cases 4-6 in Table II). As shown in the 2nd row of Table II, the variation ranges of interconnect width and spacing are almost doubled.

TABLE II
PARAMETER SPACE SPECIFICATION AND TEST CASES FOR ALGORITHM 4.

	w/w_0	l/l_0	$T-T_0$
parameter space $\mathbb{S}$	$[1,50]$	$[1,30]$	$[0,100]$
case 4	10	6	20
case 5	30	18	60
case 6	40	24	80

First, we segment the variation ranges of both w and l into 2 uniform intervals (and here we do not segment the range of T), generating 4 smaller parameter spaces. Then we use 10 frequency points to construct $\mathcal{X}$ and extract the dominant eigenspace of $\mathcal{X}\mathcal{X}^H$ to construct an order-20 parameterized ROM. In Fig. 6, the resulting ROMs' transfer functions are

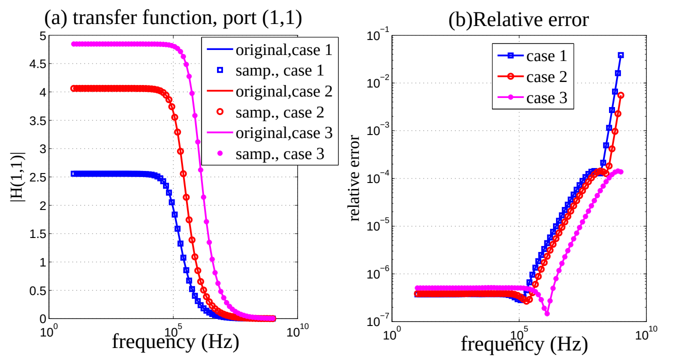


Fig. 5. MOR result by the sampling scheme modified from [15] for cases 1-3, with ROM size=20. (a) The transfer functions of the obtained ROMs are also indistinguishable from their original curves. (b) The relative errors are several orders of magnitude larger than those from our Algorithm 3.

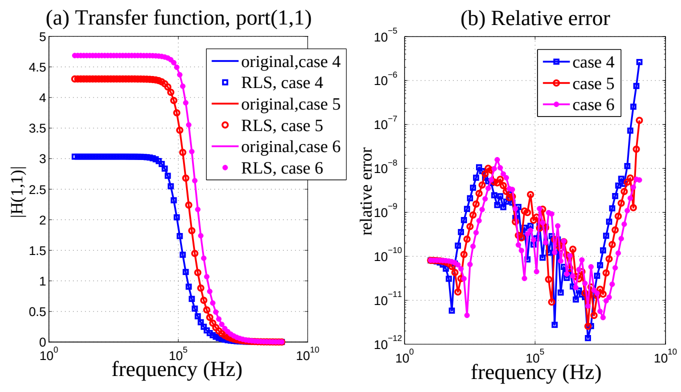


Fig. 6. Fully PMOR results by our Algorithm 4 for the large-variation cases 4-6, with ROM size=20. (a) The ROMs' transfer functions overlap with their original ones. (b) The maximum relative errors of cases 4-6 are about 10^{-5}, 10^{-7} and 10^{-8}, respectively.

still indistinguishable from the ones from the original models. For all of the three test cases (cases 4-6 in Table II), the relative errors are all below 10^{-7} when the frequency is below 10^{8}Hz.

VI. Conclusion

This paper has proposed a PMOR scheme for fully parameterized systems with possibly parameter-dependent input/output matrices. Instead of using moment matching or balanced truncation, an error minimization procedure is proposed to construct the projection matrix. Such a procedure is reformulated as a recursive least square problem and then efficiently solved. Since the optimization can eliminate much redundant information and is independent of the number of parameters, our approach can construct high-accuracy models with small ROM size for systems with large numbers of parameters. Additionally, for systems with large parameter variations, parameter space segmentation can help improve the accuracy. Our algorithms have been tested by a set of multi-parameter fully parameterized models, and have obtained both high accuracy and small ROM sizes.

References

[1] L. Daniel and J. White, "Automatic generation of geometrically parameterized reduced order models for integrated spiral RF-inductors," in *Proc. Intl. Workshop on Behavior Modeling and Simulation*. San Jose, CA, September 2003.

[2] L. Daniel, C. S. Ong, S. C. Low, K. H. Lee, and J. White, "A multi-parameter moment matching model reduction approach for generating geometrically parameterized interconnect performance models," *IEEE Trans. Computer Aided Design*, vol. 23, no. 5, pp. 678–693, May 2004.

[3] B. N. Bond and L. Daniel, "Parameterized model order reduction of nonlinear dynamical systems," in *Proc. Intl. Conf. Computer-Aided Design*. San Jose, CA, Nov 2005, pp. 487–494.

[4] ——, "A piecewise-linear moment-matching approach to parameterized model-order reduction for highly nonlinear systems," *IEEE Trans. Computer Aided Design*, vol. 26, no. 12, pp. 2116 – 2129, Dec 2007.

[5] N. Mi, S. X.-D. Tan, Y. Cai, and X. Hong, "Fast variational analysis of on-chip power grids by stochastic extended Krylov subspace method," *IEEE Trans. Computer Aided Design*, vol. 27, no. 11, pp. 1996–2006, Nov 2008.

[6] X. Li, P. Li, and L. Pileggi, "Parameterized interconnect order reduction with explicit-and-implicit multi-parameter moment matching for inter/intra-die variations," in *Proc. Intl. Conf. Computer Aided Design*, 2005, pp. 806–812.

[7] S. Pullela, N. Menezes, and L. T. Pileggi, "Moment-sensitivity-based wire sizing for skew reduction in on-chip clock nets," *IEEE Trans. Computer Aided Design*, vol. 16, no. 2, pp. 210–215, Feb 1997.

[8] Y. Liu, L. T. Pileggi, and A. J. Strojwas, "Model order reduction of RCL interconnect including variational analysis," in *Proc. Design Automation Conference*. New Orleans, Louisiana, June 1999, pp. 201–206.

[9] T. Moselhy and L. Daniel, "Variation-aware interconnect extraction using statistical moment preserving model order reduction," in *Proc. Design, Automation and Test in Europe*. Dresden, March 2010, pp. 453–458.

[10] J. F. Villena and L. M. Silveira, "SPARE - a scalable algorithm for passive, structure preserving, parameter-aware model order reduction," *IEEE Trans. Computer Aided Design*, vol. 29, no. 6, pp. 925–938, Jun 2010.

[11] Y. Li, Z. Bai, Y. Su, and X. Zeng, "Parameterized model order reduction via a two-directional Arnoldi process," in *Proc. Intl. Conf. Computer-Aided Design*. San Jose, CA, Nov 2007, pp. 868–873.

[12] J. R. Phillips, L. Daniel, and L. M. Silveira, "Guaranteed passive balancing transformations for model order reduction," *IEEE Trans. Computer Aided Design*, vol. 22, no. 8, pp. 1–15, Aug 2003.

[13] P. Heydari and M. Pedram, "Model reduction of variable-geometry interconnects using variational spectrally-weighted balanced truncation," in *Proc. Intl. Conf. Computer-Aided Deisgn*. San Jose, CA, Nov 2001, pp. 586–591.

[14] ——, "Model-order reduction using variational balanced truncation with spectral shaping," *IEEE Trans. Circuits and Systems-I: Regular Papers*, vol. 53, no. 4, pp. 879–891, April 2006.

[15] J. R. Phillips, "Variational interconnect analysis via PMTBR," in *Proc. Intl. Conf. Computer Aided Design*, 2004, pp. 872–879.

[16] K. C. Sou, A. Megretski, and L. Daniel, "A quasi-convex optimization approach to parameterized model order reduction," in *Proc. Deisgn Automation Conference*, Jun 2005, pp. 933–938.

[17] ——, "A quasi-convex optimization approach to parameterized model order reduction," *IEEE Trans. Computer Aided Design*, vol. 27, no. 3, pp. 456–469, Mar 2008.

[18] N. Dong and J. Roychowdhury, "General-purpose nonlinear model order reduction based on piecewise polynomial representations," *IEEE Trans. Computer Aided Design*, vol. 27, no. 2, pp. 249–261, Feb 2008.

[19] J. Roychowdhury, "Reduced-order modelling of time-varying systems," *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, no. 10, pp. 1273–1288, Oct 1999.

[20] P. Li and L. T. Pileggi, "Compact reduced-order modeling of weakly nonlinear analog and RF circuits," *IEEE Trans. Computer Aided Design*, vol. 23, no. 2, pp. 184–203, Feb 2005.

[21] ——, "Compact model order reduction of weakly nonlinear systems," in *Proc. Design Automation Conference*, Jun 2003, pp. 472–477.

[22] J. R. Phillips, "Projection-based approaches for model reduction of weakly nonlinear, time-varying systems," *IEEE Trans. Computer Aided Design*, vol. 22, no. 2, pp. 171–187, Feb 2003.

[23] I. M. Elfadel and D. L. Ling, "A block rational Arnoldi algorithm for multipoint passive model order reduction of multiport RLC networks," in *Proc. Intl. Conf. Computer Aided Design*, Nov 1997, pp. 66–71.

[24] A. Odabasioglu, M. Celik, and L. Pileggi, "PRIMA: passive and reduced-order interconnect macromodeling algorithm," *IEEE Trans. Computer Aided Design*, vol. 17, no. 8, pp. 645–654, Aug 1998.

[25] G. Golub and C. V. Loan, *Matrix Computations, 2nd Edition*. Baltimore, MD: Johns Hopkins University Press, 1989.