Model Reduction and Simulation of Nonlinear Circuits via Tensor Decomposition

The MIT Faculty has made this article openly available. Please share how this access benefits you. Your story matters.

| Citation          | Haotian Liu, Luca Daniel, and Ngai Wong. “Model Reduction and Simulation of Nonlinear Circuits via Tensor Decomposition.” IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 34, no. 7 (July 2015): 1059–1069. |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| As Published      | http://dx.doi.org/10.1109/TCAD.2015.2409272                                                                                                                                                       |
| Publisher         | Institute of Electrical and Electronics Engineers (IEEE)                                                                                                                                            |
| Version           | Author's final manuscript                                                                                                                                                                          |
| Accessed          | Sun Dec 09 03:50:38 EST 2018                                                                                                                                                                       |
| Citable Link      | http://hdl.handle.net/1721.1/102475                                                                                                                                                              |
| Terms of Use      | Creative Commons Attribution-Noncommercial-Share Alike                                                                                                                                             |
| Detailed Terms    | http://creativecommons.org/licenses/by-nc-sa/4.0/                                                                                                                                                 |
Model Reduction and Simulation of Nonlinear Circuits via Tensor Decomposition

Haotian Liu, Student Member, IEEE, Luca Daniel, Member, IEEE, and Ngai Wong, Member, IEEE

Abstract—Model order reduction of nonlinear circuits (especially highly nonlinear circuits), has always been a theoretically and numerically challenging task. In this paper we utilize tensors (namely, a higher order generalization of matrices) to develop a tensor-based nonlinear model order reduction (TNMOR) algorithm for the efficient simulation of nonlinear circuits. Unlike existing nonlinear model order reduction methods, in TNMOR high-order nonlinearities are captured using tensors, followed by decomposition and reduction to a compact tensor-based reduced-order model. Therefore, TNMOR completely avoids the dense reduced-order system matrices, which in turn allows faster simulation and a smaller memory requirement if relatively low-rank approximations of these tensors exist. Numerical experiments on transient and periodic steady-state analyses confirm the superior accuracy and efficiency of TNMOR, particularly in highly nonlinear scenarios.

Keywords—Tensor, nonlinear model order reduction, reduced-order model

I. INTRODUCTION

The complexity and reliability of modern VLSI chips rely heavily on the effective simulation and verification of circuits during the design phase. In particular, mixed-signal and radio-frequency (RF) modules are critical and often hard to analyze due to their intrinsic nonlinearities and their large problem sizes. Consequently, nonlinear model order reduction becomes necessary in the electronic design automation flow. The goal of nonlinear model order reduction is to find a reduced-order model that simulates fast and yet still captures the input-output behavior of the original system accurately.

Compared to the mature model order reduction methods in linear time-invariant systems [1]–[4], nonlinear model order reduction is much more challenging. Several projection-based methods have been developed in the last decade. In [5], [6], the nonlinear system is expanded into a series of cascaded linear subsystems, whereby the outputs from the low-order subsystems serve as inputs to the higher-order ones. Then, existing projection-based linear model order reduction methods, e.g., [1], [3], can be applied to these linear subsystems recursively. We refer to the method in [5], [6] as the “standard projection” approach. Nonetheless, in this method, the dimension of the resulting reduced-order model grows exponentially with respect to the orders of the subsystems. Moreover, to reduce the size of Arnoldi starting vectors, lower order projection subspaces are used to approximate the column spaces of the higher order system matrices. Consequently, approximation errors in lower order subspaces can easily propagate and accumulate in higher order subsystems.

To tackle this accuracy issue, a more compact nonlinear model order reduction scheme, called NORM, is proposed in [7] where each explicit moment of high-order Volterra transfer functions is matched. For weakly nonlinear circuits, NORM exhibits an extraordinary improvement in accuracy over the standard projection approach since lower order approximations are completely skipped. The resulting reduced-order model tends to be more compact as the sizes of lower order reduced subsystems will not carry forward to higher order ones. However, this approach still needs to build the reduced but dense system matrices whose dimensions grow exponentially as the order increases. This limits the practicality of NORM as simulation of small but dense problems is sometimes even slower than simulating the large but sparse original system.

To overcome the curse of dimensionality, rather than treating the exponentially growing system matrices as 2-dimensional matrices, their nature should be recognized. To this end, tensors, as high dimensional generalization of matrices, can be utilized. In recent years, there has been a strong trend toward the investigation of tensors and their low-rank approximation [8]–[13], due to their high dimensional nature ideal for complex problem characterization and their efficient compact representation ideal for large scale data analyses. Therefore, it is natural to characterize circuit nonlinearities by tensors whereby the tensor structure can be exploited to reduce the original nonlinear system.

In this paper, we propose a tensor-based nonlinear model order reduction (TNMOR) scheme for typical circuits with a few nonlinear components. The work is a variation of the Volterra series-based projection methods [5]–[7]. The nonlinear system is modeled by a truncated Volterra series up to a certain high order. However, in the proposed method, the higher order system matrices are modeled by high-order tensors, so that the high dimensional data can be approximated by the sum of only a few outer products of vectors via the canonical tensor decomposition [8], [12], [14]. Next, the projection spaces are generated by matching the moments of each subsystem, in terms of those decomposed vectors. Finally, the reduced-order model is represented in the canonical tensor decomposition form, where the sparsity of the high dimensional system parameters is exploited.

H. Liu and N. Wong are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong SAR (email: {htliu, nwong}@eee.hku.hk).

L. Daniel is with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, USA (email: luca@mit.edu).

Copyright © 2015 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending an email to pubs-permissions@ieee.org.
matrices is preserved after TNMOR.

The main contribution of this work is that unlike previous approaches, simulation of the TNMOR-produced reduced-order model completely avoids the overhead of solving high-order dense system matrices. This truly allows the simulation to exploit the acceleration brought about by nonlinear model order reduction. We remark that the utilization of TNMOR depends on the existence of low-rank approximations of these high-order tensors, which are generally available for circuits with a few nonlinear components. Moreover, the size of the reduced-order model depends only on the tensor rank and the order of moments being matched for each system matrix. In other words, it will not grow exponentially as the order of subsystems increases, which enables nonlinear model order reduction of highly nonlinear circuits not amenable before.

The paper is organized as follows. Section II reviews the backgrounds of Volterra series, existing nonlinear model order reduction approaches, as well as tensors and tensor decomposition. After that, Section III presents the tensor-based modeling of nonlinear systems. The proposed TNMOR is described in Section IV and simulation of the TNMOR-reduced model is discussed in Section V. Numerical examples are given in Section VI. Finally, Section VII draws the conclusion.

II. BACKGROUND AND RELATED WORK

A. Volterra subsystems

We consider a nonlinear multi-input multi-output (MIMO) time-invariant circuit modeled by the differential-algebraic equation (DAE)

$$\frac{d}{dt} \left[ q(x(t)) \right] + f(x(t)) = Bu(t), \quad y(t) = L^T x(t), \quad (1)$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^l$ are the state and input vectors, respectively; $q(\cdot)$ and $f(\cdot)$ are the nonlinear capacitance and conductance functions extracted from the modified nodal analysis (MNA); $B$ and $L$ are the input and output matrices, respectively. The nonlinear system can be expanded under a perturbation around its equilibrium point $x_0$ by the Taylor expansion

$$\frac{d}{dt} \left[ C_1 x + C_2 (x \otimes x) + C_3 (x \otimes x \otimes x) + \cdots \right] + G_1 x$$

$$+ G_2 (x \otimes x) + G_3 (x \otimes x \otimes x) + \cdots = Bu, \quad (2)$$

where $\otimes$ denotes the Kronecker product and we will use the shorthand $x^\oplus = x \otimes x \otimes x$ etc. for the Kronecker powers throughout the paper. The conductance and capacitance matrices are given by

$$G_i = \left. \frac{1}{i!} \frac{\partial^i f}{\partial x^i} \right|_{x=x_0} \in \mathbb{R}^{n \times n^i}, C_i = \left. \frac{1}{i!} \frac{\partial^i q}{\partial x^i} \right|_{x=x_0} \in \mathbb{R}^{n \times n^i}. \quad (3)$$

By Volterra theory and variational analysis [15, 16], the solution $x$ to (2) is approximated with the Volterra series

$$x(t) = x_1(t) + x_2(t) + x_3(t) + \cdots, \quad \text{where } x_i(t) \text{ is the response to each of the following Volterra subsystems}$$

$$\frac{d}{dt} \left[ C_{1,i} x + G_{1,i} x \right] + G_2 (x \otimes x) + G_3 (x \otimes x \otimes x) + \cdots = Bu, \quad (4a)$$

$$\frac{d}{dt} \left[ C_{1,i} x + G_{1,i} x \right] = - \left[ C_{2,i} (x \otimes x) + C_3 (x \otimes x \otimes x) + \cdots \right] - G_1 x, \quad (4b)$$

$$\frac{d}{dt} \left[ C_{1,i} x + G_{1,i} x \right] = - \left[ C_{2,i} (x \otimes x) + C_3 (x \otimes x \otimes x) + \cdots \right] \frac{d}{dt} G_1 x,$$

$$+ G_2 (x \otimes x) + G_3 (x \otimes x \otimes x) + \cdots = Bu, \quad (4c)$$

and so on, where $(x_{i_1} \otimes x_{i_2} \otimes x_{i_3})_k = x_{i_1} \otimes x_{i_2} + x_{i_2} \otimes x_{i_1}$.

B. Existing projection-based nonlinear model order reduction methods

To reduce the original system (2), the standard projection approach [5, 6] treats (4) as a series of MIMO linear systems, with the right hand side of each equation serving as its actual “input”. Then, the projection-based linear model order reduction approach, e.g., [1], is applied.

Suppose up to $k_t$/th-order (viz. from 0th to $k_t$/th) moments of $x(t)$ are matched by a Krylov subspace projector $x_1 \approx V_{k_t} \hat{x}_1$, the number of columns of $V_{k_t}$ is $(k_t+1)$. After that, (4b) is recast into a concatenated, stacked descriptor system and $V_{k_t}$ is used to approximate its input by assuming

$$x_1 \approx (V_{k_t} \hat{x}_1) \approx V_{k_t} \hat{x}_1 \odot \hat{x}_1, \quad (5)$$

Consequently, the multiple Krylov starting vectors of (5) become $-G_{2V_{k_t}} \odot \hat{x}_1 \odot \hat{x}_1$ instead of $-G_{2V_{k_t}} \hat{x}_1 \odot \hat{x}_1$, therefore the dimensionality of the input is reduced to $(k_t+1)^2 l^2$ from $n^2$. If up to $k_t$/th order moments of $x(t)$ are preserved, (4b) can be reduced by another projection $x_2 \approx V_{k_t} \hat{x}_2$ to a smaller linear system with $(k_t+1)(k_t+1)^2 l^2$ inputs. Similarly, higher order projectors $V_{k_t}, V_{k_t}, \ldots$ are obtained by iteratively reducing the remaining subsystems in (4).

Suppose we have $N$ linear subsystems in (4) and $k = k_1 = k_2 = \cdots = k_N$ order of moments are matched in each subsystem, the standard projection approach will result in a reduced-order model with size $O(k^{2N-1} N)$. Thus, in practical circuit reduction examples, $k_1, k_2, \ldots$ should be relatively small (otherwise the size of the reduced system may even exceed $n$ quickly), which could hamper the accuracy of the reduced-order model.

Instead of regarding (4) as a set of linear equations, NOR-M [7] derives frequency-domain high-order nonlinear Volterra
transfer functions $H_2(s_1, s_2)$, $H_3(s_1, s_2, s_3)$, etc. associated to the subsystems in (4). These transfer functions are expanded into multivariate polynomials of $s_1, s_2, \ldots$ such that the coefficients (moments) can be explicitly matched. In NORM, the size of an $N$th-order reduced-order model is in $O(k^{N+1}N)$ if $k = k_1 = \ldots = k_N$.

In both aforementioned methods, the final step consists of replacing the original nonlinear system (2) by a smaller system via the transformations

$$
\tilde{x} = V^T x, \quad \tilde{B} = V^T B, \quad \tilde{L} = V^T L,
$$

where $i = 1, \ldots, N$ and $V = \text{orth}[V_1, V_2, V_3, \ldots]$ is the orthogonal projector. Suppose $q$ is the size of the reduced state, $G_i$ and $C_i$ will be dense matrices with $O(q^{i+1})$ entries, despite the sparsity of $G_i$ and $C_i$. To store these dense matrices, the memory space required grows exponentially.

C. Tensors and tensor decomposition

Some tensor basics are reviewed here, while more tensor properties and decompositions can be found in [9].

1) Tensors: A $d$th-order tensor is a $d$-way array defined by

$$
\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}.
$$

For example, Fig. 1(a) depicts a 3rd-order $3 \times 4 \times 2$ tensor. In particular, scalars, vectors and matrices can be regarded as 0th-order, 1st-order and 2nd-order tensors, respectively.

Matricization is a process that unfolds or flattens a tensor into a 2nd-order matrix. The $k$-mode matricization of $\mathcal{A}$ is denoted as $\mathcal{A}(k)$ and is defined by $\mathcal{A}(k) = \mathcal{A}(1) \otimes \mathcal{A}(2) \otimes \cdots \otimes \mathcal{A}(k)$, where $\mathcal{A}(i)$ is the $i$th-order reduced-order model in (7).

A conceptual explanation of $k$-mode product is to multiply each $k$th-direction “vector fiber” in $\mathcal{A}$ by the matrix $U$. An illustration of the multiplication to a 3rd-order tensor is shown in Fig. 1(b).

The “Khariti-Rao product” is the “matching columnwise” Kronecker product. The Khariti-Rao product of matrices $A = [a_{11}, a_{22}, \ldots, a_{kk}] \in \mathbb{R}^{n_1 \times k}$ and $B = [b_{11}, b_{22}, \ldots, b_{kk}] \in \mathbb{R}^{n_2 \times k}$ is defined as $A \otimes B = [a_{11} \otimes b_{11}, a_{22} \otimes b_{22}, \ldots, a_{kk} \otimes b_{kk}] \in \mathbb{R}^{n_1 n_2 \times k}$. If $A$ and $B$ are column vectors, $A \otimes B = A \otimes B$. And if $A$ and $B$ are row vectors, $A \otimes B$ becomes the Hadamard product (viz. element-by-element product) of the two rows.

3) Rank-1 tensors and canonical decomposition: A rank-1 tensor of dimension $d$ can be written as the outer product of $d$ vectors

$$
\mathcal{A} = a^{(1)} \circ a^{(2)} \circ \cdots \circ a^{(d)}, \quad a^{(k)} \in \mathbb{R}^{n_k},
$$

where $\circ$ denotes the outer product. Its element $\mathcal{A}_{i_1 i_2 \cdots i_d} = a^{(1)}_{i_1} \cdot a^{(2)}_{i_2} \cdots a^{(d)}_{i_d}$, where $a^{(k)}_{i_k}$ is the $i_k$th entry of vector $a^{(k)}$.

The CANDECOMP/PARAFAC (CP) decomposition [8], [9], [14], [17] approximates a tensor $\mathcal{A}$ by a finite sum of rank-1 tensors, which can be written by

$$
\mathcal{A} \approx \sum_{r=1}^R a^{(1), r} \circ a^{(2), r} \circ \cdots \circ a^{(d), r}, \quad a^{(k), r} \in \mathbb{R}^{n_k},
$$

where $R \in \mathbb{N}$. Concisely, using the factor matrices $\mathcal{A}(k) \triangleq [a^{(1), k}, a^{(2), k}, \ldots, a^{(k), k}] \in \mathbb{R}^{n_k \times R}$, the right-hand side of the CP (10) can be expressed by the notation $\mathcal{A} \approx \sum_{r=1}^R \mathcal{A}(k)^{r}$. Moreover, it is worth mentioning that the $k$th-mode matricization of $\mathcal{A}$ could be reconstructed by these factor matrices

$$
\mathcal{A}(k) \approx \mathcal{A}(k)^{(1)} \circ \cdots \circ \mathcal{A}(k)^{(R)} \otimes \mathcal{A}(k)^{(1)} \otimes \cdots \otimes \mathcal{A}(k)^{(1)},
$$

The rank of the tensor $\mathcal{A}$, rank($\mathcal{A}$), is the minimum value of $R$ in the exact decomposition (10). A rank-$R$ approximation of a 3rd-order tensor is shown in Fig. 3.

Several methods have been developed to compute the CP decomposition, for example, the alternating least squares (ALS) [8], [17] (as well as many of its derivatives) or the optimization methods such as CPOPT [11]. Most of them solve the optimization problem of minimizing the Frobenius norm

1 The complexity refers to the single-point expansion algorithm of NORM. The multi-point version of NORM would have a lower complexity.

2 We denote tensors by calligraphic letters, e.g., $\mathcal{A}$ and $G$. 

Fig. 1. (a) A tensor $\mathcal{A} \in \mathbb{R}^{3 \times 4 \times 2}$. (b) Illustration of $\mathcal{A} \times U_1 \times U_2 \times U_3$.
ALS is several times faster than CPOPT in general. However, of the difference between the original tensor and its rank-R approximation

\[
\min f(A^{(1)}, \ldots, A^{(d)}) \triangleq \frac{1}{2} \left\| A - \left[ A^{(1)}, \ldots, A^{(d)} \right] \right\|_F^2.
\]

The ALS algorithm iteratively optimizes one factor matrix \(A^{(i)}\) at a time, by holding all other factors fixed and solving the linear least square problem

\[
\min f(A^{(1)}, \ldots, A^{(d)})
\]

for the updated \(A^{(i)}\). Alternatively, CPOPT calculates the gradient of the objective function \(f\) in (12) and uses the generic nonlinear conjugate gradient method to optimize (12). For both ALS and CPOPT, the rank \(R\) should be prescribed and is fixed during the computation. It is reported in [11] that the computational complexities for both ALS and CPOPT to approximate an \(N\)th-order tensor \(A \in \mathbb{R}^{n_1 \times \cdots \times n_N}\) are \(O(NQR)\) per iteration, where \(Q = \prod_{i=1}^{N} n_i\). It is also mentioned in [11] that ALS is several times faster than CPOPT in general. However CPOPT shows an “essentially perfect” accuracy compared with ALS. A review of different CP methods also can be found in [9].

\[\text{III. TENSOR-FORM MODELING OF NONLINEAR SYSTEMS}\]

To begin with, we give an equivalent tensor-based modeling of the nonlinear system (2). Recall the definitions of \(G_i\) and \(C_i\) in (3), it is readily found that these coefficient matrices are respectively 1-mode matricizations of \((i+1)\)th-order tensors \(G_i\) and \(C_i\),

\[
G_i, C_i \in \mathbb{R}^{n \times \cdots \times n},
\]

where the elements \((G_i)_{j_1 j_2 \cdots j_i}\) and \((C_i)_{j_1 j_2 \cdots j_i}\) are coefficients of the \(\prod_{k=1}^{i} x_{j_k}\) term in \(G_i\) and \(C_i\), respectively. For instance, \(G_2\) is an \(n \times n^2\) matrix while \(G_3\) is a 3rd-order \(n \times n \times n\) tensor, i.e., \(G_3\) is the 1-mode matricization of \(G_2\).

According to Proposition 3.7 in [13], the Kronecker products in (2) can be represented by the tensor mode multiplication via

\[
G_1(x^{(\odot)}) = G_1 x_1 x_2 x_3 x_i x_{i+1} x_T,
\]

and

\[
C_1(x^{(\odot)}) = C_1 x_1 x_T x_3 T x_i x_{i+1} x_T. \]

Therefore, (2) is equivalent to

\[
\frac{d}{dt} \left[ G_1(x_T) + C_2 x_T \right] + C_2 x_T x_3 T \cdots x_i T \cdots x_{i+1} x_T + \cdots = Bu.
\]

The key to the tensor-form modeling is to pre-decompose these high dimensional tensors via CP. In practical circuit systems, in spite of the growing dimensionality, high-order nonlinear coefficients \(G_i\) and \(C_i\) (\(G_1\) and \(C_1\)) are almost always sparse. Therefore it is advantageous to make a rank-\(R\) CP approximation of \(G_i\) or \(C_i\) for a relatively small \(R\). In other words, we can use a few rank-1 tensors to express \(G_i\) and \(C_i\) by

\[
G_i \approx \left[ G_i^{(1)}, \ldots, G_i^{(i+1)} \right] = \sum_{r=1}^{r_g} g_{i, r}^{(1)} \cdots g_{i, r}^{(i+1)},
\]

\[
C_i \approx \left[ C_i^{(1)}, \ldots, C_i^{(i+1)} \right] = \sum_{r=1}^{r_c} c_{i, r}^{(1)} \cdots c_{i, r}^{(i+1)},
\]

where \(i = 2, \ldots, N, r_g\) and \(r_c\) are the tensor ranks of \(G_i\) and \(C_i\), respectively, \(g_{i, r}^{(k)} \in \mathbb{R}^{n_1 \times \cdots \times n_N}\), \(G_i^{(k)} = [g_{i, 1}^{(k)}, \ldots, g_{i, r_g}^{(k)}] \in \mathbb{R}^{n \times r_g}, \) \(C_i^{(k)} = [c_{i, 1}^{(k)}, \ldots, c_{i, r_c}^{(k)}] \in \mathbb{R}^{n \times r_c}, \) for \(k = 1, \ldots, i + 1\). It should be noticed that different permutations of indices can result in the same polynomial term. For example, term \(x_1 x_2\) can be represented by any combination of \(x_3 x_1 x_2 + (1 - \alpha) x_2 x_1\). Therefore, the high-order tensors are not unique for a specific nonlinear system and the consequent low-rank approximations (16) could be very different. Nonetheless, we use the tensors with minimum nonzero entries in our implementation, as they tend to be sparser such that lower rank approximations are usually available.

Using the CP structure (16), the original nonlinear system (15) can be approximated by absorbing tensor products of \(x\) into the factor matrices

\[
\frac{d}{dt} \left[ C_1 x_T + C_2^{(1)} (x_T C_2^{(2)} \circ x_T C_2^{(3)}) + \cdots \right] + G_1 x_T = Bu.
\]

Applying (11), the 1-mode matricization of (17) is simply

\[
\frac{d}{dt} \left[ C_1 + C_2^{(1)} (x_T C_2^{(2)} \circ x_T C_2^{(3)}) + \cdots \right] + G_1 + C_2^{(1)} (x_T C_3^{(2)} \circ x_T C_3^{(3)}) \cdots = Bu,
\]

and its corresponding 1-mode matricized Volterra subsystems are given by

\[
\frac{d}{dt} [C_1 x_1] + G_1 x_1 = Bu,
\]

\[
\frac{d}{dt} [C_1 x_2] + G_1 x_2 = - \frac{d}{dt} \left[ C_2^{(1)} (x_T C_2^{(3)} \circ x_T C_2^{(2)}) \right] + G_2^{(1)} (x_T C_2^{(4)} \circ x_T C_2^{(3)}) + \cdots = Bu,
\]

and so on.
IV. TNMOR

Without loss of generality, we start from (18) and (19) to derive TNMOR. It can be observed that (19) is a series of linear systems where \( x_1 \) is solved by the first equation with input \( v \), \( x_2 \) is the solution to the second linear system with the input dependent on \( x_1 \), and similarly \( x_3 \) is solved in the third system with its input dependent on \( x_1 \) and \( x_2 \).

Similar to [1], [5]–[7], the frequency-domain transfer function of (19a) is given by

\[
H_1(s) = (sG_1^{-1}C_1 + I)^{-1}G_1^{-1}B \hat{=} (-sA_1 + I)^{-1}B_1. \tag{20}
\]

To match up to \( k \)-th order of (20), its projection space \( V_{k_1} \) is the Krylov subspace of \( K_{k_1+1}(A_1, B_1) \) if \( H_1(s) \) is expanded around the origin, where \( K_{m}(A, p) = \text{span}\{p, Ap, \ldots, A^{m-1}p\} \). The Krylov subspace can be efficiently calculated by the block Arnoldi iteration [1].

The 2nd-order subsystem (19b) can be recast into

\[
\begin{bmatrix}
C_1 & -I & 0 \\
0 & 0 & x_2 \\
0 & 0 & x_2e
\end{bmatrix}
\begin{bmatrix}
\hat{x}_2 \\
\hat{x}_3
\end{bmatrix}
+
\begin{bmatrix}
G_1 & 0 \\
0 & I \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
x_2 \\
x_3 \\
x_3e
\end{bmatrix}
= \begin{bmatrix}
G_2^{(1)} & 0 \\
0 & C_2^{(1)}
\end{bmatrix}
\begin{bmatrix}
(x^T G_3^{(2)} \otimes x^T G_2^{(2)})^T \\
(x^T G_3^{(3)} \otimes x^T G_2^{(2)})^T
\end{bmatrix}
\begin{bmatrix}
\hat{x}_2 \\
\hat{x}_3
\end{bmatrix}.
\tag{21}
\]

Consequently, (21) could be treated as a linear system with \( r_{g,2} + r_{c,2} \) inputs and its transfer function reads

\[
H_2(s) = \left(s \begin{bmatrix}
G_1^{-1}C_1 & -G_1^{-1} \\
0 & 0
\end{bmatrix} + I\right)^{-1}
\begin{bmatrix}
-G_1^{-1}G_2^{(1)} & 0 \\
0 & -C_2^{(1)}
\end{bmatrix}
\hat{=} (-sA_2 + I)^{-1}B_2. \tag{22}
\]

Suppose \( k_2 \)-th order of \( H_2(s) \) are matched, its Krylov subspace is obtained by \( K_{k_2+1}(A_2, B_2) \). Thus, we have \( V_{k_2} \) to be the first \( n \) rows of \( K_{k_2+1}(A_2, B_2) \) (noticing \( H_2(s) \) is a \( 2n \times 1 \) vector).

For the 3rd-order subsystem (19c), similarly, it could be represented by another linear system

\[
\begin{bmatrix}
C_1 & -I & 0 \\
0 & 0 & x_3 \\
0 & 0 & x_3e
\end{bmatrix}
\begin{bmatrix}
\hat{x}_3 \\
\hat{x}_4
\end{bmatrix}
+
\begin{bmatrix}
G_1 & 0 \\
0 & I \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
x_3 \\
x_4 \\
x_3e
\end{bmatrix}
= \begin{bmatrix}
G_3^{(1)} & 0 \\
0 & C_3^{(1)}
\end{bmatrix}
\begin{bmatrix}
(x^T G_4^{(4)} \otimes x^T G_3^{(4)} \otimes x^T G_2^{(2)})^T \\
(x^T G_4^{(4)} \otimes x^T G_3^{(5)} \otimes x^T G_2^{(2)})^T
\end{bmatrix}
\begin{bmatrix}
\hat{x}_3 \\
\hat{x}_4
\end{bmatrix}.
\tag{23}
\]

with \( r_{g,3} + r_{c,3} + r_{g,2} + r_{c,2} \) inputs. Moreover, the 3rd-order transfer function is given by

\[
H_3(s) = \left(s \begin{bmatrix}
G_1^{-1}C_1 & -G_1^{-1} \\
0 & 0
\end{bmatrix} + I\right)^{-1}
\begin{bmatrix}
-G_1^{-1}G_3^{(1)} & 0 \\
0 & -C_3^{(1)}
\end{bmatrix}
\hat{=} (-sA_2 + I)^{-1}[B_3 B_2]. \tag{24}
\]

Consequently, the Krylov subspace of the 3rd-order subsystem should be \( K_{k_3+1}(A_2, [B_3 B_2]) = K_{k_3+1}(A_2, B_3) \) if the number of moments being matched is \( k_3 \). However, it is readily seen that if \( k_3 \leq k_2 \), we have \( K_{k_3+1}(A_2, B_2) \subseteq K_{k_2+1}(A_2, B_2) \). Since \( K_{k_3+1}(A_2, B_2) \) has already been obtained from \( H_2(s) \), \( K_{k_3+1}(A_2, B_2) \) does not need to be recomputed. Therefore, only \( K_{k_3+1}(A_2, B_2) \) should be counted at this stage and we denote its first \( n \) rows to be \( V_{k_3} \).

Higher order linear transfer functions can be obtained in a similar way. The \( n \)-th order projector \( V_{k_n} \) is the first \( n \) rows of \( K_{k_h+1}(A_2, B_1) \), where \( B_i = \begin{bmatrix} -G_1^{-1}G_i^{(1)} & 0 \\ 0 & -C_i^{(1)} \end{bmatrix} \).

The reducing projector for the nonlinear system is the orthogonal basis of all \( V_{k_i} \), denoted by \( V = \text{orth}(V_{k_1}, V_{k_2}, \ldots) \). The size of an \( N \)-th order reduced-order model is \( O(k_1l + k_2(r_{g,2} + r_{c,2}) + k_3(r_{g,3} + r_{c,3}) + \cdots + k_N(r_{g,N} + r_{c,N})) = O(Nkr) \), where \( k = \max\{k_1, \ldots, k_N\} \) and \( r = \max\{r_{g,2} + r_{c,2}, \ldots, r_{g,N} + r_{c,N}\} \). Comparing with \( O(k^{2N-1}N^2) \) in the standard projection approach and \( O(k^{N+1}N^2) \) in NORM, a slimmer reduced-order model can be achieved if low-rank CP of the tensors are available.

Finally, the tensor-based reduced-order model is given by the following projection

\[
\hat{x} = V^T x, \quad \tilde{B} = V^TB, \quad \tilde{L} = V^TL, \quad \tilde{G}_i = \begin{bmatrix} G_i^{(1)} & \ldots & G_i^{(k_i+1)} \end{bmatrix} = \begin{bmatrix} V^TG_i^{(1)}, \ldots, V^TG_i^{(k_i+1)}\end{bmatrix}, \quad \tilde{G}_i = \begin{bmatrix} C_i^{(1)} & \ldots & C_i^{(k_i+1)} \end{bmatrix} = \begin{bmatrix} V^TC_i^{(1)}, \ldots, V^TC_i^{(k_i+1)}\end{bmatrix}, \tag{25}
\]

which looks similar to (6) except for the tensor CP structure. By utilizing this structure, only the \( G_i^{(k_i)} \) and \( C_i^{(k_i)} \) matrices need to be stored, and simulation of the reduced-order model can be significantly speeded up as will be seen in the next section, as long as low-rank CP approximations of \( \tilde{G}_i \) and \( \tilde{C}_i \) are available. Algorithm 1 summarizes the TNMOR assuming a DC expansion point.

The computational complexity of TNMOR is dominated by the CP decompositions of \( \tilde{G}_i \) and \( \tilde{C}_i \). Any CP method can be used to extract the rank-1 factors. We use CPOPT to compute the CP in TNMOR. Although CPOPT is not as remarkably fast as ALS, we find in practice that CPOPT can always achieve a better accuracy under the same rank. The computational cost for CPOPT to optimize a rank-\( r_{g,N} \) approximation of \( G_N \) is \( O(Nnr_{g,N}^2) \) per iteration. By contrast, the costs for NORM and the standard projection method to calculate the Krylov starting vectors for an \( N \)-th order subsystem are \( O(k^{3N}n^2r_{g,N}) \). It should be noticed that the CP decompositions only need to be done once and the resulting reduced CP structure can be used in different on-going simulations.

Another key issue is how to heuristically prescribe/estimate the rank for each tensor. It is readily found that accurate low-rank CP approximations of the high dimensional tensors is critical to the proposed method. Although it will be shown in Section V that the size of the reduced-order model and the time complexity for its simulation are proportional to the tensor ranks \( r_{GS} \) and \( r_{CS} \), the efficiency of the proposed
Algorithm 1 TMNOR Algorithm

Input: \( N, G_1, C_i, B, L, k_1, k_N \leq k_{N-1} \leq \cdots \leq k_1 \)
Output: \( G_1, C_i, \tilde{G}_i, \tilde{C}_i, B, L, i = 2, \ldots, N \)

1: for \( i = 2 \) to \( N \) do
2: \( \tilde{G}_i = G_i; \tilde{C}_i = C_i; \)
3: \( [G_1^{(1)}, \ldots, G_i^{(i+1)}] = CP(\tilde{G}_i); \)
4: \( [C_1^{(1)}, \ldots, C_i^{(i+1)}] = CP(\tilde{C}_i); \)
5: end for
6: \( A_1 = -G_1^{-1}C_1; B_1 = G_1^{-1}B; \)
7: \( V_{k_1} = K_{k_1+1}(A_1, B_1); \)
8: for \( i = 2 \) to \( N \) do
9: \( A_2 = \begin{bmatrix} G_1^{-1}C_1 & -G_1^{-1} \\ 0 & 0 \end{bmatrix}; B_i = \begin{bmatrix} -G_1^{-1}C_1^{(1)} & 0 \\ 0 & -C_i^{(1)} \end{bmatrix}; \)
10: \( V_i = K_{k_1+1}(A_2, B_i); \)
11: end for
12: \( V = \text{orth}(V_{k_1}, V_2, \ldots, V_{k_N}); \)
13: \( \tilde{G}_1 = V^T G_1 V; \tilde{C}_1 = V^T C_1 V; \tilde{B} = V^T B; \tilde{L} = V^T L; \)
14: for \( i = 2 \) to \( N \) do
15: \( \tilde{G}_i = [V^T G_i^{(1)}, \ldots, V^T G_i^{(i+1)}]; \)
16: \( \tilde{C}_i = [V^T C_i^{(1)}, \ldots, V^T C_i^{(i+1)}]; \)
17: end for

A. Function evaluation

Rewrite (2) for the reduced-order model
\[
 f(\tilde{x}) \triangleq \tilde{x} - \tilde{G}_1 \tilde{x} - \tilde{G}_2 \tilde{x} \otimes \tilde{G}_3 \tilde{x} \otimes \cdots + \tilde{B} u, \quad (26)
\]
where \( f(\tilde{x}) \) is the function to be evaluated in the simulation. Using the reduced-order model in (25), the equivalent 1-mode matricization of (26) is
\[
 f(\tilde{x}) = -\tilde{G}_1 \tilde{x} - \tilde{G}_2^{(1)} \left( \tilde{x}^T \tilde{G}_3^{(3)} \otimes \tilde{x}^T \tilde{G}_2^{(2)} \right)^T \\
 - \tilde{G}_3^{(1)} \left( \tilde{x}^T \tilde{G}_3^{(4)} \otimes \tilde{x}^T \tilde{G}_3^{(3)} \otimes \tilde{x}^T \tilde{G}_3^{(2)} \right)^T - \cdots + \tilde{B} u. \quad (27)
\]
It should be noticed that all \( \tilde{x} \tilde{G}_3^{(k)} \) and \( \tilde{x} \tilde{C}_3^{(k)} \) are row vectors, therefore \( \otimes \) corresponds to the element-by-element multiplication between matrices. If up to the \( N \)-th order nonlinearity is included and the size of the reduced-order model is \( q \), the time complexity for evaluating (27) is \( O(N^2 q r_g) \) where \( r_g = \max\{r_g,2, \ldots, r_g,N\} \), while it is in \( O(q^{N+2}) \) for the model (6) used in the standard projection approach and NORM.

B. Jacobian matrix

Consider the Jacobian matrix of \( f(\tilde{x}) \) which is often used in time-domain simulation
\[
 J_f(\tilde{x}) \triangleq -\tilde{G}_1 - \tilde{G}_2 (\tilde{x} \otimes I + I \otimes \tilde{x}) \\
 - \tilde{G}_3 (\tilde{x} \otimes \tilde{x} \otimes I + \tilde{x} \otimes I \otimes \tilde{x} + I \otimes \tilde{x} \otimes \tilde{x}) - \cdots. \quad (28)
\]
Its equivalent 1-mode matricization is given by
\[
 J_f(\tilde{x}) = -\tilde{G}_1 - \tilde{G}_2^{(1)} \left( \tilde{x}^T \tilde{G}_3^{(3)} \otimes \tilde{G}_2^{(2)} + \tilde{G}_2^{(3)} \otimes \tilde{x}^T \tilde{G}_2^{(2)} \right)^T \\
 - \tilde{G}_3^{(1)} \left( \tilde{x}^T \tilde{G}_3^{(4)} \otimes \tilde{x}^T \tilde{G}_3^{(3)} + \tilde{x}^T \tilde{G}_3^{(4)} \otimes \tilde{x}^T \tilde{G}_3^{(2)} \right)^T + \tilde{G}_3^{(4)} \otimes \tilde{x}^T \tilde{G}_3^{(3)} \otimes \tilde{x}^T \tilde{G}_3^{(2)} + \cdots.
\quad (29)
\]
The complexity for the standard projection method or NORM to calculate (28) is in \( O(Nq(N+q) r_g) \). For the tensor formulation, the complexity for evaluating (29) is further reduced to \( O(Nq(N + q) r_g) \).

C. Space complexity

In our proposed method, the amount of memory space for the reduced-order model is determined by the factor matrices in (25), which should be in \( O(Nq r_g) \), where \( r = \max\{r_g,2, \ldots, r_g,N\} \). For the existing standard projection approach and NORM, the storage consumption is dominated by the matrices \( G_N \) and \( C_N \) whose numbers of elements are in \( O(q^{N+1}) \).

V. SIMULATION OF THE REDUCED-ORDER MODEL

Here we show that whenever low-rank tensor approximations of \( G_i \) and \( C_i \) exist, TMNOR can help to accelerate simulation and avoids the exponential growth of the memory requirement, versus the reduced but dense models generated by the standard projection method or NORM. We describe the time complexity by means of the two key processes in circuit simulation, namely, function evaluation and calculation of the Jacobian matrices. For the ease of illustration, we assume the conductance matrix \( C_i = I \) and all higher order \( C_i = 0 \), \( i = 2, \ldots, N \). Extension to general cases is straightforward.

VI. NUMERICAL EXAMPLES

In this section, TMNOR is demonstrated and compared with the standard projection approach and NORM. All experiments are implemented in MATLAB on a desktop with an Intel i5 2500@3.3GHz CPU and 16GB RAM. To fairly present the results, all time-domain transient analyses are solved by the trapezoid discretization with fixed step sizes. In the simulations...
of the original system, the reduced-order models of the standard projection and NORM approaches, (26) and (28) are used to evaluate the functions and Jacobian matrices, while (27) and (29) are used for the tensor-based model. The CP in TNMOR is computed by the CPOPT algorithm provided in the MATLAB Tensor Toolbox [11], [12], [19].

A. A double-balanced mixer

First, we study a double-balanced mixer circuit in Fig. 4 [20], where \( V_{if}(= V_{if+} - V_{if-}) \) and \( V_{lo}(= V_{lo+} - V_{lo-}) \) are the RF and local oscillator (LO) inputs, respectively. We assume \( V_{rf} \) and \( V_{lo} \) are both sinusoidal and their frequencies are 2GHz and 200MHz, respectively. \( V_{if+} \) and \( V_{if-} \) are the intermediate-frequency (IF) outputs. The size \( n \) of the original system is 93. Firstly, we assume a relatively small \( V_{lo} \) swing so that the nonlinear system can be approximated by its 3rd-order Taylor expansion

\[
\frac{d}{dt}[C_1 x] + G_1 x + G_2 x^3 + G_3 x^3 = Bu, \tag{30}
\]

where the numbers of nonzero elements in \( G_2 \) and \( G_3 \) are 16 and 30, respectively.

Then, the standard projection approach, NORM and TNMOR are applied to (30). It should be noticed that all methods are expanded at the frequencies 2GHz and 200MHz. The size of the reduced-order model, the order of the moments \( k_i \) matched in each subsystem, and the CPU times for each model order reduction method are listed in Table I. It can be noticed that due to the curse of dimensionality, the standard projection approach generates a larger (and denser) reduced-order model but fewer orders of moments are persevered. TNMOR takes 1.3s to optimize a best rank-6 \((r_{g,2} = 6)\) approximation of \( G_2 \) with the error \( \|G_2 - \hat{G}_2\|_F = 4.2 \times 10^{-4} \), and 3.1s for a best rank-9 \((r_{g,3} = 9)\) approximation of \( G_3 \) with an error \( 1.2 \times 10^{-4} \).

A transient simulation from 0 to \( T = 25\)ns with a \( \Delta t = 5\)ps step size is performed on each reduced-order model. The runtimes and overall errors of different methods are summarized in Table II. The overall error is defined as the relative error between the vector of \( V_{if} \) at subsequent timesteps \( V = [V_{if}(0), V_{if}(\Delta t), \ldots, V_{if}(T)]^T \) of the reduced-order models and the corresponding vector computed with the full model, i.e., \( \|V - \hat{V}\|_2 \). The first 7.5ns of the transient waveforms of \( V_{if} \) and their relative errors are plotted in Figs. 5(a) and 5(b), respectively.

Next, the periodic steady-state analyses of different models are achieved by a shooting Newton method-based periodic steady-state simulator. The CPU times and the overall errors of the frequency responses between 0 to 4GHz are listed in Table III. The relative errors of the frequency responses are plotted in Fig. 5(c).

It is shown in Tables II and III that although TNMOR generates a larger reduced-order model than NORM (39 versus 30), its transient and periodic steady-state analyses are faster due to the efficient algorithm of function and Jacobian evaluations. In contrast, simulations of the dense reduced-order models generated by NORM and the standard projection approach are much slower than the original large but sparse system, indicating that these methods are impractical for strongly nonlinear systems (3rd-order nonlinearity in this example). This is mainly because though both (26) and (28) have been used in the simulations, the Kronecker powers in (26) and (28) of the original system never have to be explicitly computed due to the sparsity of the original \( G_i \) and \( C_i \) matrices. Moreover, comparing to NORM, the proposed method provides a competitive accuracy as can be seen in Fig. 5.

Practically, mixers are often modeled by periodic time-varying systems due to the existence of large \( V_{lo} \) signals [5]-
with transfer functions \( \hat{G}_i(s) \) of the periodic steady-state simulations. (\( \text{SP} \) is the acronym for “standard projection” \cite{5, 6}.)

Fortunately, TNMOR can be easily extended to periodic time-varying systems as well. Following \cite{5}–\cite{7}, suppose all higher order \( \bar{C}_i = 0 \), all the system matrices in (19) become \( T_{\text{lo}} \)-periodic where \( T_{\text{lo}} \) is the period of \( V_{\text{lo}} \). Next, a uniform backward-Euler discretization over sampling points \( [t_1, t_2, \ldots, t_M] \) where \( t_i = \frac{i}{M} T_{\text{lo}} \) is applied on each linear time-varying system in (19), resulting a set of LTV systems with transfer functions \( \hat{H}_i(s) \)

\[
[\hat{J}_i + s\hat{C}_i] \hat{H}_i(s) = \hat{B}_i,
\]

(31)

where

\[
\hat{H}_i(s) = \begin{bmatrix} H_i^T(t_1, s) & H_i^T(t_2, s) & \ldots & H_i^T(t_M, s) \end{bmatrix}^T,
\]

\[
\hat{B}_1 = \begin{bmatrix} B_i^T(t_1) & B_i^T(t_2) & \ldots & B_i^T(t_M) \end{bmatrix}^T,
\]

\[
\hat{G}_1 = \begin{bmatrix} G_1(t_1) \\ G_1(t_2) \\ \vdots \\ G_1(t_M) \end{bmatrix},
\]

(32)

and

\[
\hat{B}_i = -\hat{G}_i^{(1)} = -\begin{bmatrix} G_i^{(1)}(t_1) \\ G_i^{(1)}(t_2) \\ \vdots \\ G_i^{(1)}(t_M) \end{bmatrix},
\]

(34)

We omit the detailed derivation as it is straightforward. Similarly, the projector is obtained by collocating the moments of each order subsystem in (31). It should be noticed that the factor matrices at each sampling point can be computed individually via CP decompositions. Therefore the computational complexity is proportional to \( M \).

Next, we reformulate the mixer by a 3rd-order nonlinear time-varying system under a large signal \( V_{\text{lo}} \). The 930-variable system is expanded around \( M = 10 \) operating points. This time, TNMOR, multi-point NORM (NORM-mp) and the standard projection method are expanded at 2GHz. The sizes of reduced-order models and CPU times are listed in Table IV. The CPU time of TNMOR is mainly spent on sequential CP decompositions, which could be further reduced if multi-core parallelization is enabled.

These reduced-order models are simulated for 3rd-order intermodulation tests. The LO and RF frequencies are fixed.
while we sweep the amplitude of the sinusoidal RF input from 1mV to 10mV. Fig. 6 shows the 3rd-order intermodulation product (IM3) results of the original system and reduced-order models. The CPU times are summarized in Table V, where they are written as $a \pm b$ where $a$ is the average value and $b$ is the sample standard deviation. From Fig. 6, good agreement can be observed for the reduced-order models generated by NORM-mp and TNMOR. NORM-mp achieves a smaller size because it only preserves the values of nonlinear transfer functions at specific points, which is particularly useful for matching high-order distortions. Meanwhile, TNMOR still demonstrates comparable accuracy and better efficiency due to the benefit of the tensor framework.

### B. A biochemical reaction system

The second example is a sparse biochemical reaction system model adapted from [22]. The system is generated by a random 2nd-order polynomial system with a $\frac{1}{1+x}$ function

$$\frac{d}{dt}x + G_1 x + G_2 x^2 + e_1 \frac{10}{1+x} = Bu,$$

where $G_1 \in \mathbb{R}^{200 \times 200}$, $G_2 \in \mathbb{R}^{200 \times 200^d}$, $B \in \mathbb{R}^{200 \times 3}$ and $e_1 \in \mathbb{R}^{200 \times 1} = [1, 0, \ldots, 0]^T$. It should be noticed that both $G_1$ and $B$ are dense random matrices and the eigenvalues of $G_1$ are randomly distributed on $(0, 3]$ so that the nonlinear system is stable at the origin. $G_2$ is a sparse random matrix with 48 nonzero entries. $\frac{1}{1+x}$ is expended by the Taylor series $\frac{1}{1+x} \approx 1-x+\frac{x^2}{2}-\frac{x^3}{3}$ and we control the inputs to guarantee $|x| < 1$ during the simulations.

The three nonlinear model order reduction approaches are applied on the 3rd-order polynomial system to generate the reduced-order models. We match the moments at the origin in all approaches. The sizes of the reduced-order models, the orders of the moments in each subsystem and the CPU times for the reduction have been listed in Table VI. TNMOR optimizes a rank-20 ($r_{G_2} = 20$) approximation of $G_2$ in 11s with a relative error 0.03. The unique nonzero element in $G_3$ is $-10x_1^3$, therefore its CP can be obtained immediately with the rank $r_{G_3} = 1$.

We feed these reduced-order models by 10 sets of sinusoidal inputs for transient simulations with the same time period and the same step size. These sinusoidal inputs are under different frequencies. The CPU times and errors have been summarized in Table VII. The CPU times further confirm that for sparse systems, TNMOR can utilize the sparsity while the structure cannot be kept in NORM or the standard projection approach.

### C. A 2-D pulse-narrowing transmission line

It is reported in [23] that linear lossy transmission line would cause the wave dispersion effect of the input pulse which could be avoided if certain nonlinear capacitors are introduced. We consider a nonlinear pulse-narrowing transmission line shown in Fig. 7(a) [23], [24]. There are two pulse inputs injected at the two corners of the transmission line. We are interested in the voltage at the center of the shaded edge. The length and width of the transmission line are 20cm and 10cm, respectively. It is uniformly partitioned into a $20 \times 10$ grid, with each node shown by the lumped circuit in Fig. 7(b). The nodes at the shaded edge of the transmission line are characterized by the nonlinear state equation $C_0 \dot{V}_{i,j}(t) = \sum_{i=1}^{N} V_{i,j} - V_{i,j}(1 + \sum_{i=1}^{N}(bV_{i,j})^N) - V_{i,j}/r_g(1 + \sum_{i=1}^{N}(bV_{i,j})^N)$ where $C_0 = 1 \mu F$, $r_g = 10 K \Omega$ and $b = -0.9$, while the nonlinear coefficient $b = 0$ elsewhere. In Fig. 7(b), $L = 1 \mu H$ and $R = 0.12$.

Using MNA, we end up with a system with 570 state variables. First, we approximate the original system by the 3rd-order Taylor expansion, i.e., up to $G_3$. There are 78 nonzero elements in each order $G_i$, $i = 1, 2, \ldots$. In this case, all the moments are matched at the origin as well. Again, the sizes of the reduced-order models, the orders of the moments in each subsystem and the CPU times for the reduction are listed in Table VIII. Due to the exponential growth of the size of

| Method | $k_1$ | $k_2$ | $k_3$ | CPU time | size of ROM |
|--------|---|---|---|---|---|
| standard projection [5], [6] | 1 | 0 | 0 | 0.02s | 39 |
| NORM [7] | 2 | 1 | 0 | 1.2s | 90x |
| TNMOR | 2 | 2 | 0 | 11s | 39 |

| Method | $k_1$ | $k_2$ | $k_3$ | CPU time | size of ROM |
|--------|---|---|---|---|---|
| standard projection [5], [6] | 1 | 0 | 0 | 0.02s | 39 |
| NORM [7] | 2 | 1 | 0 | 1.2s | 90x |
| TNMOR | 2 | 2 | 0 | 11s | 39 |
on these vectors to generate a compact reduced-order model. Projection-based model order reduction is employed just a few vectors obtained from the canonical tensor decom-
tors such that these nonlinearities can be represented by high dimensional 

scheme called TNMOR is proposed. The high-order nonlinearities in the system are characterized by high dimensional tensors, thereby easing memory requirement and speeding up computation via exploitation of data structures. Examples have been shown to demonstrate the efficiency of TNMOR over existing nonlinear model order reduction algorithms.

VII. CONCLUSION

In this paper, a tensor-based nonlinear model order reduction scheme called TNMOR is proposed. The high-order nonlinearities in the system are characterized by high dimensional tensors such that these nonlinearities can be represented by just a few vectors obtained from the canonical tensor decomposition. Projection-based model order reduction is employed on these vectors to generate a compact reduced-order model under the tensor framework. The key feature of TNMOR is that it preserves the inherent sparsity through low-rank tensors, thereby easing memory requirement and speeding up computation via exploitation of data structures. Examples have been shown to demonstrate the efficiency of TNMOR over existing nonlinear model order reduction algorithms.

ACKNOWLEDGEMENT

This work was supported in part by the Hong Kong Research Grants Council under General Research Fund (GRF) Projects 718213E and 17208514, the University Research Committee of The University of Hong Kong, and the MIT-China MISTI program. The authors are also thankful to Prof. Ivan Oseledets of Skoltech and Mr. Zheng Zhang of MIT for their valuable comments and advice.

REFERENCES

[1] A. Odabasioglu, M. Celik, and L. Pileggi, “Prima: passive reduced-
order interconnect macromodeling algorithm,” Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, vol. 17, no. 8, pp. 645–654, Aug 1998.

[2] J. Phillips, L. Daniel, and L. Silveira, “Guaranteed passive balancing transformations for model order reduction,” Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, vol. 22, no. 8, pp. 1027–1041, Aug 2003.

[3] P. Feldmann and R. Freund, “Efficient linear circuit analysis by node approximation via the Lanczos process,” Computer-Aided Design of integrated Circuits and Systems, IEEE Transactions on, vol. 14, no. 5, pp. 639–649, May 1995.

[4] B. Bond and L. Daniel, “Guaranteed stable projection-based model reduction for indefinite and unstable linear systems,” in Computer-Aided Design, 2008. ICCAD 2008. IEEE/ACM International Conference on, Nov 2008, pp. 728–735.
[5] J. Phillips, “Automated extraction of nonlinear circuit macromodels,” in *Custom Integrated Circuits Conference*, 2000. CICC. Proceedings of the IEEE 2000, 2000, pp. 451–454.

[6] J. Roychowdhury, “Reduced-order modeling of time-varying systems,” *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 46, no. 10, pp. 1273–1288, Oct 1999.

[7] P. Li and L. Pileggi, “Compact reduced-order modeling of weakly nonlinear analog and RF circuits,” *IEEE Trans. Comput-Aided Design Integr. Circuits Syst.*, vol. 23, no. 2, pp. 184–203, Feb. 2005.

[8] J. Carroll and J.-J. Chang, “Analysis of individual differences in multidimensional scaling via an n-way generalization of “Eckart-Young” decomposition,” *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[9] T. Kolda and B. Bader, “Tensor decompositions and applications,” *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[10] I. Oseledets, “Tensor-train decomposition,” *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2295–2317, 2011.

[11] E. Acar, D. M. Dunlavy, and T. G. Kolda, “A scalable optimization approach for fitting canonical tensor decompositions,” *Journal of Chemometrics*, vol. 25, no. 2, pp. 67–86, February 2011.

[12] B. W. Bader and T. G. Kolda, “Efficient MATLAB computations with sparse and factored tensors,” *SIAM Journal on Scientific Computing*, vol. 30, no. 1, pp. 205–231, December 2007.

[13] T. G. Kolda, “Multilinear operators for higher-order decompositions,” Sandia National Laboratories, Tech. Rep. SAND2006-2081, April 2006. [Online]. Available: http://www.osti.gov/scitech/biblio/923081

[14] B. W. Bader and T. G. Kolda, “Algorithm 862: MATLAB tensor classes for fast algorithm prototyping,” *ACM Transactions on Mathematical Software*, vol. 32, no. 4, pp. 635–653, December 2006.

[15] W. Rugh, *Nonlinear System Theory – The Volterra-Wiener Approach*. Baltimore, MD: Johns Hopkins Univ. Press, 1981.

[16] E. Bedrosian and S. O. Rice, “The output properties of Volterra systems (nonlinear systems with memory) driven by harmonic and Gaussian inputs,” *Proc. IEE*, vol. 59, no. 12, pp. 1688–1707, Dec. 1971.

[17] R. A. Harshman, “Foundations of the PARAFAC procedure: Models and conditions for an ‘explanatory’ multi-modal factor analysis,” *UCLA Working Papers in Phonetics*, vol. 16, no. 1, p. 84, 1970.

[18] J. Håstad, “Tensor rank is NP-complete,” *J. Algorithms*, vol. 11, no. 4, pp. 644–654, Dec. 1990.

[19] B. W. Bader, T. G. Kolda et al., “MATLAB Tensor Toolbox Version 2.5.” Available online, January 2012. [Online]. Available: http://www.sandia.gov/~tgkolda/TensorToolbox/

[20] W. Dong and P. Li, “A parallel harmonic-balance approach to steady-state and envelope-following simulation of driven and autonomous circuits,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 28, no. 4, pp. 490–501, April 2009.

[21] R. Telichevsky, K. Kundert, and J. White, “Efficient AC and noise analysis of two-tone RF circuits,” in *Design Automation Conference Proceedings 1996*, 33rd, Jun 1996, pp. 292–297.

[22] C. Gu, “QLMOR: a projection-based nonlinear model order reduction approach using quadratic-linear representation of nonlinear systems,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 9, pp. 1307–1320, Sep. 2011.

[23] E. Afshari and A. Hajimiri, “Nonlinear transmission lines for pulse shaping in silicon,” *Solid-State Circuits, IEEE Journal of*, vol. 40, no. 3, pp. 744–752, 2005.

[24] B. N. Bond, “Stability-preserving model reduction for linear and nonlinear systems arising in analog circuit applications;” Ph.D. dissertation, Massachusetts Institute of Technology, Feb. 2010.