

PAPER

Fast Analysis of On-Chip Power Grid Circuits By Extended Truncated Balanced Realization Method*

Duo LI[†], *Nonmember and* Sheldon X.-D. TAN[†], *Member*

SUMMARY In this paper, we present a novel analysis approach for large on-chip power grid circuit analysis. The new approach, called *ETBR* for extended truncated balanced realization, is based on model order reduction techniques to reduce the circuit matrices before the simulation. Different from the (improved) extended Krylov subspace methods EKS/IEKS [2, 3], *ETBR* performs fast truncated balanced realization on response Gramian to reduce the original system. *ETBR* also avoids the adverse explicit moment representation of the input signals. Instead, it uses spectrum representation in frequency domain for input signals by fast Fourier transformation. The proposed method is very amenable for threading-based parallel computing, as the response Gramian is computed in a Monte-Carlo-like sampling style and each sampling can be computed in parallel. This contrasts with all the Krylov subspace based methods like the EKS method, where moments have to be computed in a sequential order. *ETBR* is also more flexible for different types of input sources and can better capture the high frequency contents than EKS, and this leads to more accurate results especially for fast changing input signals. Experimental results on a number of large networks (up to one million nodes) show that, given the same order of the reduced model, *ETBR* is indeed more accurate than the EKS method especially for input sources rich in high-frequency components. If parallel computing is explored, *ETBR* can be an order of magnitude faster than the EKS/IEKS method.

key words: *power grid analysis, model order reduction, truncated balanced realization*

1. Introduction

Reliable on-chip power delivery is one of the major challenges for 90nm and below VLSI technology. This situation becomes worse as technology continues to scale owing to the several reasons: First, technology scaling results in decreased interconnect width and increased interconnect resistance in a power supply network. Second, increased device density leads to increased supply current density on a chip. Third, a higher clock frequency gives rise to more significant inductance effect. At the same time, supply voltage continues to decrease, which results in a decreased noise margin for signal transition, and makes transistor more vulnerable to supply voltage degradation. So efficient verification of power integrity becomes critical for final design closure.

Many research works have been done on efficient simulation of on-chip power grid networks. Methods such as multigrid-like [4, 5], hierarchical [3, 6], partition-based [7], fast iterative [8] and random walk based [9, 10] approaches help improve scalability of power grid network analysis. Another approach to fast power grid analysis is based on so-called extended Krylov subspace based methods (EKS) [2, 3]. In EKS methods, both a power grid system and its input signals are used to reduce the original circuits before the simulation. Due to efficiency of Krylov subspace based reduction techniques, EKS/IEKS can deal with very large power grid circuits. But EKS method also suffers several shortcomings. First, the methods need to represent the input signals in the Taylor expansion form or the moment form with respect to complex frequency variable s . This can lead to less accurate results when the input signals are fast changing waveforms with many spike-like shapes. Such high-frequency bearing input waveforms cannot be represented accurately using the moment form owing to the well-known problems in explicit moment matching methods [11, 12]. Second, EKS is based on the Krylov subspace method to reduce the circuit matrices. The Krylov subspace approach leads to localized accuracy due to moment-matching property. Multiple-point moment matching will result in larger reduced systems, which degrades the simulation efficiency.

In this paper, we propose a novel model order reduction based simulation approach. This approach, called *ETBR* for extended truncated balanced realization, is based on the similar idea of the EKS method, where both a system and its input signals are used to reduce the original circuit matrices. But different from the (improved) extended Krylov subspace methods, EKS/IEKS [2, 3], *ETBR* performs singular value decomposition (SVD) on response Gramian to reduce the original system while with the similar computation costs of EKS. *ETBR* is based on a more accurate reduction framework: truncated balanced realization, which was shown to be more accurate than Krylov subspace method used in EKS method.

The proposed method is very amenable for threading-based parallel computing, as the response Gramian, which is used to construct the projection matrix, is computed in a Monte-Carlo-like sampling style and each sampling can be computed in parallel. This contrasts with all the Krylov subspace based methods like the EKS method, where moments have to be com-

[†]Duo Li and Sheldon X.-D. Tan are with Department of Electrical Engineering, University of California, Riverside, CA 92521, USA (e-mail: {dli, stan}@ee.ucr.edu)

*Some preliminary results of this paper appeared *Proc. Proc. Design, Automation and Test in Europe (DATE08)* [1].

This work is supported in part by NSF award under grant No. CCF-0448534, in part by UC MICRO awards #07-105 and #08-12 via Cadence Design System Inc.

puted in a sequential order. The feature is important as the multi-core architectures and multi-core computing are becoming commonplace [13, 14]. ETBR can naturally exploit task-level threading-oriented parallelism based on multicore architectures to significantly boost the simulation performance.

ETBR also avoids the explicit moment representation of the input signals, which have well-known numerical problems in the past. Instead, it uses spectrum representation of input signals by fast Fourier transformation. As a result, ETBR is much more flexible for different types of input sources and can better capture the high frequency contents than EKS and this leads to more accurate results for fast changing input signals. Experimental results, on a number of large RLC networks up to one million nodes, show that ETBR is indeed more accurate than the EKS/IEKS method especially for current sources rich in high-frequency components. ETBR also shows similar computational costs of EKS but smaller memory footprint in a single CPU, but is much faster than EKS when parallelism is explored.

The rest of this paper is organized as follows: Section 2 presents the power grid models used in the paper. Section 3 presents the new ETBR method. We also review the standard and fast balanced truncation reduction methods in this section. Section 4 presents the experimental results and Section 5 concludes this paper.

2. Power grid network models

The power grid networks in this paper are modeled as RC networks with known time-variant current sources, which can be obtained by gate level logic simulations of the circuits. Fig. 1 shows the power grid models used in this paper. For a power grid, some nodes having known voltage are modeled as constant voltage sources. For C4 power grids, the known voltage nodes can be internal nodes inside the power grid. Given the current source vector, $u(t)$, the node voltages can be obtained by solving the following differential equations, which is formulated using modified nodal analysis (MNA) approach,

$$Gv(t) + C \frac{dv(t)}{dt} = Bu(t) \quad (1)$$

where $G \in R^{n \times n}$ is the conductance matrix, $C \in R^{n \times n}$ is the matrix resulting from storage elements. $v(t)$ is the vector of time-varying node voltages and branch currents of voltage sources. $u(t)$ is the vector of independent power sources, and B is the input selector matrix.

3. New extended balanced truncation method: ETBR

In this paper, we propose an extended truncated balanced realization method, called ETBR, for efficient simulation of power grid networks. The new method

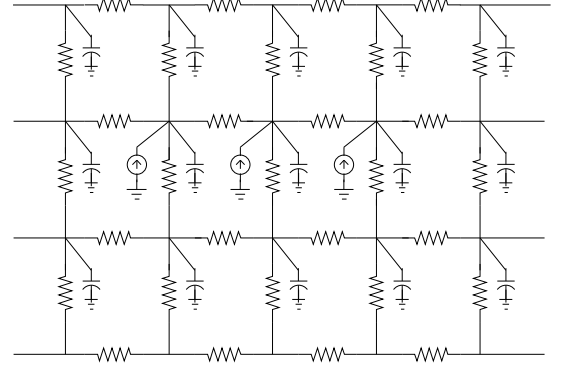


Fig. 1 The power grid model used.

features two improvements over existing approaches. First, the input signals are represented in its spectrum form in frequency domain directly by fast Fourier transformation. Second, fast balanced truncation method is used to perform the reduction, which has global accuracy [15, 16].

In the following, we first review the balanced truncation method and then the fast Gramian computation method.

3.1 Review of standard TBR

Given a system in a standard state-space form

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \quad (2)$$

where $A \in R^{n \times n}$, $B \in R^{n \times p}$, $C \in R^{p \times n}$, $y(t)$, $u(t) \in R^p$. The controllable and observable Grammians are the unique symmetric positive definite solutions to the Lyapunov equations.

$$\begin{aligned} AX + XA^T + BB^T &= 0 \\ A^TY + YA + C^TC &= 0 \end{aligned} \quad (3)$$

Since the eigenvalues of the product XY are invariant under similarity transformation, we can perform a similarity transformation ($A_b = T^{-1}AT$, $B_b = T^{-1}B$, $C_b = CT$) to diagonalize the product XY such that

$$T^{-1}XYT = \Sigma = \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_n^2) \quad (4)$$

where the Hankel singular values of the system (σ_k), are arranged in a descending order. If we partition the matrices as

$$\begin{bmatrix} W_1^T \\ W_2^T \end{bmatrix} XY \begin{bmatrix} V_1 & V_2 \end{bmatrix} = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \quad (5)$$

where $\Sigma_1 = \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_r^2)$ are the first r largest eigenvalues of Gramian product XY and W_1 and V_1 are corresponding eigenvectors. A reduced model can be obtained as follows

$$\begin{aligned} \dot{x}(t) &= A_r x(t) + B_r u(t) \\ y(t) &= C_r x(t) \end{aligned} \quad (6)$$

where $A_r = W_1^T A V_1$, $B_r = W_1^T B$, $C_r = C V_1$. The error in the transfer function of the order r approximation is bounded by $2 \sum_{i=r+1}^N \sigma_k$. In the TBR procedure, the computational cost is dominated by solving Lyapunov equations $O(n^3)$, which makes it too expensive to apply to integrated circuits problems and thus an efficient Gramian approximation technique is highly appreciated.

3.2 Review of fast TBR method: Poor man's TBR

Existing Gramian approximation technique, PMTBR [17], is restricted to a state-space model (2) with $A = A^T$ and $C = B^T$. This is the case for RC and RL circuits. In this symmetric case, it is easy to see that, both Gramians are equal and are obtained by solving the Lyapunov equation

$$AX + XA^T + BB^T = 0 \quad (7)$$

Since X is symmetric, it is orthogonally diagonalizable, i.e., there exists $T^{-1} = T^T$ such that $T^T X T = \Sigma$. Then, we have

$$T^T X X T = (T^T X T)(T^T X T) = (\Sigma)^2 \quad (8)$$

which means, in this symmetric case, the eigenspace of Gramian product XX is exactly the eigenspace of each X and we only need to find the dominant invariant subspace of an approximated Gramian \hat{X} . In frequency domain, the Gramian X can also be computed from the expression

$$X = \int_{-\infty}^{+\infty} (j\omega I - A)^{-1} B B^T (j\omega I - A)^{-H} d\omega \quad (9)$$

where superscript H denotes Hermitian transpose. Let ω_k be k th sampling point. If we define

$$z_k = (j\omega_k I - A)^{-1} B \quad (10)$$

then X can be approximated as

$$\hat{X} = \sum w_k z_k z_k^H = Z W^2 Z^H \quad (11)$$

where $Z = [z_1, z_2, \dots, z_n]$. W a diagonal matrix with diagonal entries $w_{kk} = \sqrt{w_k}$. w_k comes from a specific numerical quadrature method. Since \hat{X} is symmetric, it is orthogonally diagonalizable.

$$\hat{V}^T \hat{X} \hat{V} = \begin{bmatrix} \hat{V}_1^T \\ \hat{V}_2^T \end{bmatrix} \hat{X} \begin{bmatrix} \hat{V}_1 & \hat{V}_2 \end{bmatrix} = \begin{bmatrix} \hat{\Sigma}_1 & 0 \\ 0 & \hat{\Sigma}_2 \end{bmatrix} \quad (12)$$

where $\hat{V}^T \hat{V} = I$. \hat{V} converges to the eigenspaces of X and the dominant eigenvectors \hat{V}_1 can be used as the projection matrix in a model reduction approach ($A_r = \hat{V}_1^T A \hat{V}_1$, $B_r = \hat{V}_1^T B$).

3.3 Response Gramian and fast computation method

Follow the similar strategy of EKS method, we consider the input signals of the system into TBR based reduction framework so that efficient reduction can be done by converting an MIMO system into an SIMO system.

For a linear system in (1), we first define the frequency-domain *Response Gramian*,

$$X_r = \int_{-\infty}^{+\infty} (j\omega C + G)^{-1} B u(j\omega) u^T(j\omega) B^T (j\omega C + G)^{-H} d\omega \quad (13)$$

which is different from the Gramian concepts in the traditional TBR based reduction framework. Notice that in the new Gramian definition, the input signals $u(j\omega)$ is considered. As a results, $(j\omega C + G)^{-1} B u(j\omega)$ actually is the system response with respect to the input signal $u(j\omega)$ and resulting X_r becomes response Gramian.

To fast compute the response gramian X_r , which essential essentially one-dimensional integral with respect to the complex frequency ω . We can use Monte-Carlo-based method to estimate the numerical value as done in [18]. Specifically, let ω_k be k th sampling point over the frequency range. If we further define

$$z_k^r = (j\omega_k C + G)^{-1} B u(j\omega_k) \quad (14)$$

then \hat{X} can be computed approximately by numerical quadrature methods

$$\hat{X}_r = \sum_k w_k z_k^r z_k^{rH} = Z_r W^2 Z_r^H \quad (15)$$

where Z_r is a matrix whose columns are z_k^r and W a diagonal matrix with diagonal entries $w_{kk} = \sqrt{w_k}$. w_k comes from a specific quadrature method.

For the truncated balanced based reduction, we need to compute the eigen-composition of \hat{X}_r to obtain the projection matrix, which consists of eigen vectors of \hat{X}_r . Since the approximate Gramian \hat{X}_r is symmetric, we can obtain the project matrix by singular value decomposition of Z_r . To see this, if we have SVD of $Z_r = V_r S_r U_r^T$, then we can have the eigen decomposition of $\hat{X}_r = V_r S_r^2 V_r^T$. After this, we can reduce the original matrices into small ones and then perform the transient analysis on the reduced circuit matrices.

Also we find that weights w_k are not important for the SVD process. The weight matrix W will not change the subspace of Z_r as it simple multiplies each vector in Z_r with a constant. In our algorithm, we just simple ignore the weights and we use simple linear or logarithmic sampling methods to perform the sampling (to be discussed later).

Notice that we need frequency response of input signal $u(j\omega_k)$ in (14). This can be obtained by fast Fourier transformation on the input signals in time domain.

3.4 Extended truncated balanced realization method: ETBR

In this subsection, we give the algorithm flow of the proposed ETBR method, which is summarized in *Algorithm 1*.

Algorithm 1: Extended Truncated Balanced Realization method (ETBR)

Input: Circuit of $G, C, B, u(t)$, number of samples: q , transient simulation step interval.

Output: Node voltage responses $v(t)$ for the given simulation interval.

1. Convert all the input signals $u(t)$ into $u(s)$ using FFT.
 2. Select q frequency points s_1, s_2, \dots, s_q over the frequency range
 3. Compute $z_k^r = (s_k C + G)^{-1} B u(s_k)$
 4. Form the matrix $Z_r = [z_1^r, z_2^r, \dots, z_q^r]$
 5. Perform SVD on $Z_r, Z_r = V_r S_r U_r^T$
 6. $\hat{G} = V_r^T G V_r, \hat{C} = V_r^T C V_r, \hat{B} = V_r^T B$
 7. Simulate $(\hat{G}, \hat{C}, \hat{B}, u(t))$
 8. Obtain the original waveforms $v(t) = V_r v(\hat{t})$
-

Note that, like the EKS method, we use congruence transformation for the reduction process with orthogonal columns in the projection matrix (using Arnoldi or Arnoldi-like process), the reduced system must be stable. As far as simulation is concerned, this is good enough. If all the observable ports are also the current source nodes, i.e. $y(t) = B^T v(t)$, where $y(t)$ is the voltage vector at all observable ports, the reduced system is passive.

Compared with the existing approaches like EKS/IKES methods, ETBR shows several advantages and features. First ETBR method is much more amenable for parallel computing than EKS as each z_i^k in (14) can be computed in parallel. Thus ETBR is more efficient than EKS when the threading-based parallel computing is explored as shown in the experimental results. Second, it is more accurate over wide band frequency ranges due to the global samplings. Third, it avoids the explicit moment representation of the input signals, which can lead more accurate results than the EKS method when signals are rich in high frequency components. ETBR can deal with any type of time-domain and frequency-domain input signals. While the EKS method can only deal with input signals in piecewise linear form.

3.5 Time complexity analysis and comparison

In this subsection, we analyze the computational costs for both ETBR and EKS and compare with the EKS methods.

In ETBR, there are two major computing steps,

sampling and SVD. Let's look at the cost of each step. For sampling, we basically need to solve the (14) q times. Typically solving a $n \times n$ linear matrix takes $O(n^\beta)$ (typically, $1.1 \leq \beta \leq 1.5$ for matrix factorizations and $O(n^\alpha)$ (typically, $1 \leq \alpha \leq 1.2$) for solving (forward and backward substitutions). So the time complexity for this step is $O(qn^\beta + qn^\alpha)$. For the second step, the singular value decomposition (SVD) will take $O(nq^2)$ for a $n \times q$ matrix. Another computing cost comes from converting the input signals into the frequency spectrum form. Assume that we have m current sources, the samplings we use for the FFT is l , FFT takes $O(l \log l)$ to finish. Hence the cost associated with input signals is $O(ml \log l)$. Typically we set $l = 128$, which gives sufficient accuracy. So the total computational cost of ETBR is

$$O(qn^\beta + qn^\alpha + nq^2 + ml \log l). \quad (16)$$

If all the sampling can be computed in parallel, computing (14) will become $O(n^\beta + n^\alpha)$ assuming very small overheads incurred to manage the threads. The total computational cost of ETBR will become

$$O(n^\beta + n^\alpha + nq^2 + ml \log l). \quad (17)$$

For one-point (expanded at one frequency point) EKS, it also have two major computing costs: compute the response moments and orthonormalize them similar to the QR decomposition. For the first step, it will take one matrix factorization and q steps solving (forward and backward substitutions). The computing cost is $O(qn^\alpha + n^\beta)$, where $O(qn^\alpha)$ (typically, $1 \leq \alpha \leq 1.2$ for sparse circuits) is q step solving. The orthonormalization will take about $O(nq^2)$ to finish. Again, we need to calculate the computing cost for transforming the input signals into the moment form. It can be shown that the computing cost is $O(qk^2m)$ [3], where k is the number of piecewise segments in each current sources, and m the number of current sources. Hence the final computational cost for EKS is

$$O(qn^\alpha + n^\beta + nq^2 + qk^2m) \quad (18)$$

It can be seen that EKS will be more efficient due to smaller number of factorizations in a single CPU. But if parallel computing is allowed, ETBR become much better. But if iteration solvers are used, which are typically more fast and memory efficient than the LU-based direct solvers for RLC networks [19], both approaches will have the same computational costs in a single CPU.

3.6 Statistical point of view

The proposed method in a sense can be viewed as special SVD-based principal component analysis (PCA) method used in statistical variable reduction transformation.

For a linear dynamic system formulated in state space equations (MNA) in (1), if complex frequency s

Table 1 Test circuits

Test Ckts	#Nodes	#Sources
Ckt1	1,000	100
Ckt2	10,000	100
Ckt3	10,000	1,000
Ckt4	100,000	1,000
Ckt5	100,000	4,000
Ckt6	500,000	5,000
Ckt7	500,000	20,000
Ckt8	1,000,000	50,000
Ckt9 (RLC)	6,000	100
Ckt10 (RLC)	250,000	100

is a vector of random variables with uniform distribution in the frequency domain. Then the state responses $V(s) = (G + sC)^{-1}Bu(s)$ become random variables in frequency domain. Its covariance matrix can be computed as

$$E\{V(s)V(s)^T\} = X_r \quad (19)$$

where $E\{x\}$ stands for computing the mean of random variable x . X_r is defined in (13). The response Gramian essentially can be viewed as the covariance matrix associated with state responses. ETBR procedure performs the principal component analysis transformation of the mentioned random process with a uniform distribution.

4. Experimental results

The proposed *ETBR* algorithm has been implemented using MATLAB 7.0 and tested on an Intel Xeon 3.0GHz dual CPU workstation with 2GB memory and an Intel quad-core 3.0GHz CPU workstation with 16GB memory. All the test circuits are randomly generated RC or RLC power grid networks up to one million nodes (R on the order of Ω , C on the order of pF and L on the order of pH), as shown in Table 1. Efficient matrix computations benefit from sparse matrix structure and a parser implemented by Python.

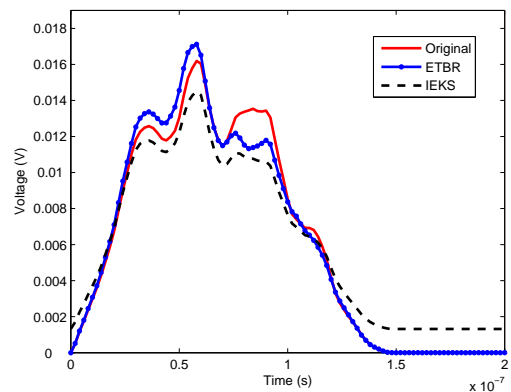
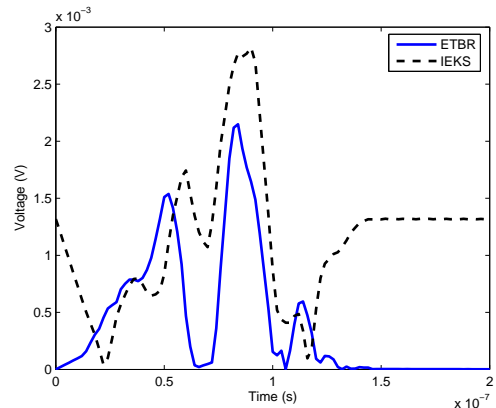
To solve circuits with one million nodes in MATLAB, an external linear solver package UMFPACK [4] is used, which is linked with MATLAB using MATLAB mexFunction. For ETBR, we use a non-LU-decomposition solver in UMFPACK. While for EKS, the LU decomposition solver is used. The reason why we choose different solver for them is ETBR only solves one column in the right hand side, so LU decomposition may cost too much and cannot be reused in ETBR. While in EKS, LU decomposition can be reused to solve several columns in the right hand side, number of columns depending on the selected reduced order, so doing LU decomposition is an efficient way in EKS. We remark that the selection of solvers is the best for both ETBR and EKS. The comparison is more fair for them.

4.1 Comparison with the EKS method

In sequel, we will compare our ETBR with IEKS [3], first in accuracy and then in CPU times. In all the test cases, to make a fair comparison, the reduction order q is set to 6 for IEKS and the number of frequency

samples used for ETBR is also set to 6. Note that for the RLC circuits Ckt1-Ckt8, the results are collected on an Intel dual CPU workstation with 2GB memory, and for the RLC circuits Ckt9 and Ckt10, the results are collected on an Intel quad-core workstation with 16GB memory.

Fig. 2 shows the simulation results of ETBR and IEKS at the 200th node of Ckt2. The simulation errors compared with SPICE results are shown in Fig. 3. One of the input signal waveforms in both time domain and frequency domain is as shown in Fig. 4. Through Fig. 3, we can see that ETBR is more accurate than IEKS over the entire simulation time.


Fig. 2 Transient waveform at the 200th node of Ckt2.

Fig. 3 The simulation errors of ETBR and IEKS of Ckt2.

In the second testing case, we change the input signals so that they can have more fast changing spikes as shown in Fig. 7(a). In other words, current sources are rich in high-frequency components.

We find that ETBR's results are much better than EKS's as shown in Fig. 5. From the simulation errors comparison in Fig. 6, we can see that ETBR is almost $3\times$ more accurate than IEKS (the maximum

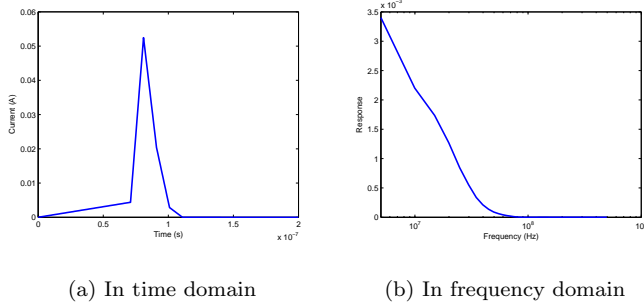


Fig. 4 Transient waveform at the 5th current source of Ckt2.

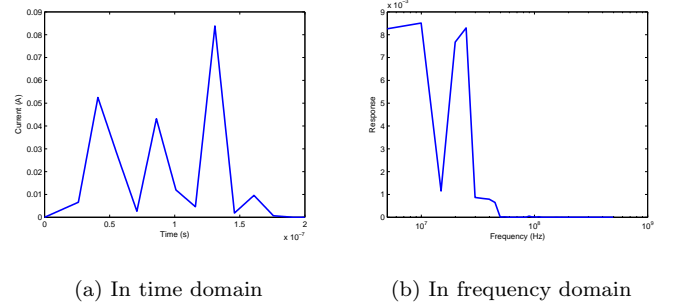


Fig. 7 The transient waveform at the 5th current source of Ckt2.

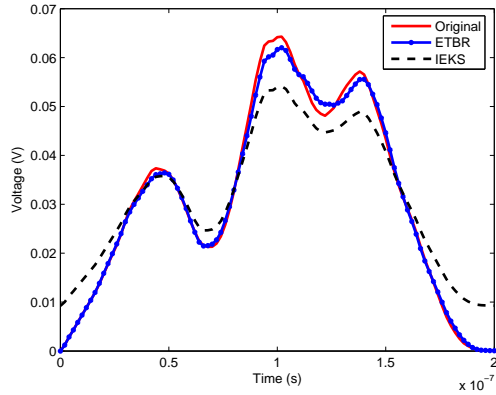


Fig. 5 Transient waveform at the 200th node of Ckt2 with fast changing inputs.

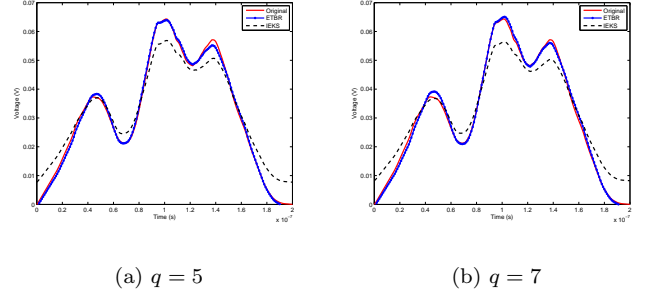


Fig. 8 Transient waveform at the 200th node of Ckt2 with fast changing inputs.

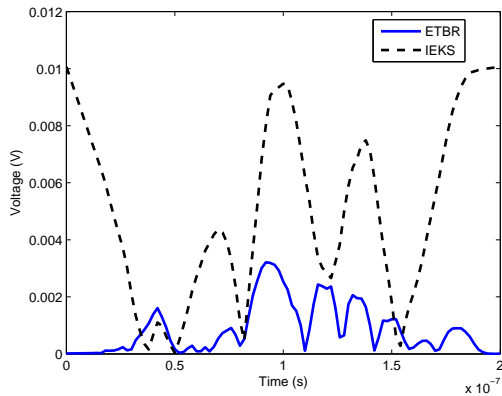


Fig. 6 The simulation errors of ETBR and IEKS on Ckt2 with fast changing inputs.

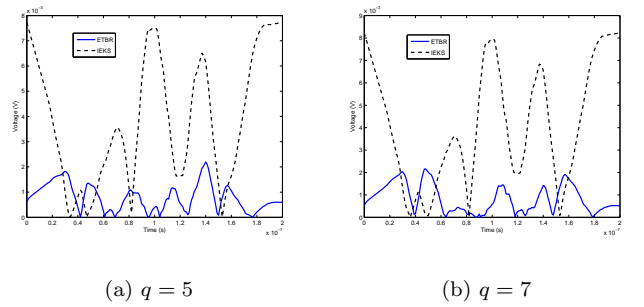


Fig. 9 The simulation errors of ETBR and IEKS on Ckt2 with fast changing inputs.

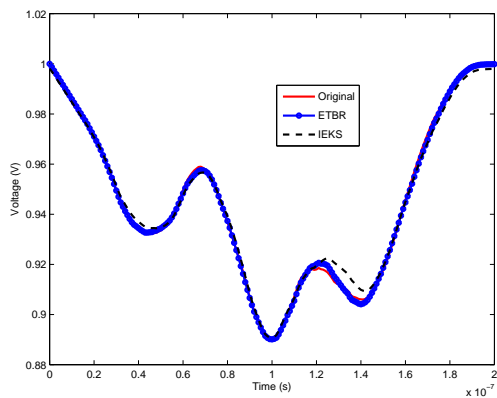
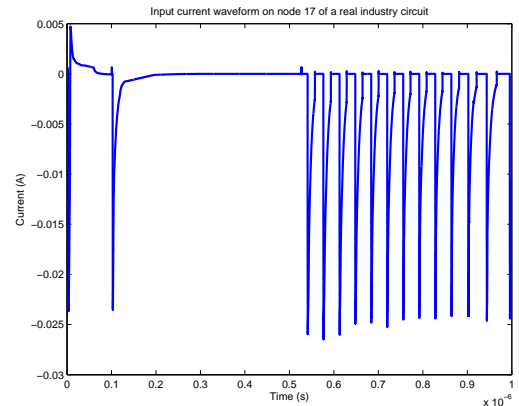
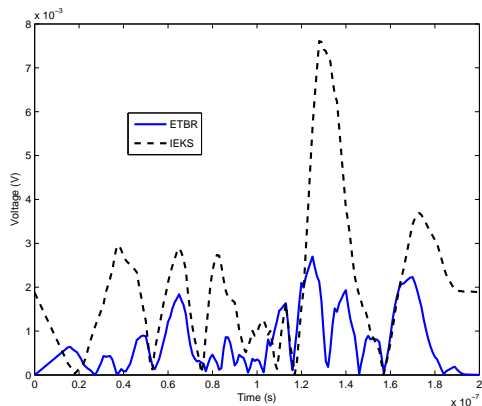
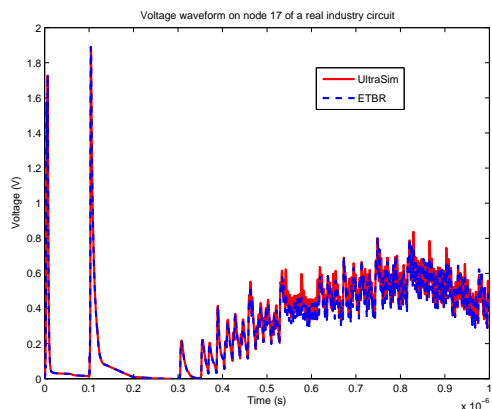
error: ETBR 0.003 vs IEKS 0.01). This is not a surprise for us if we notice that the input signals shown in Fig. 7(b) have much more high frequency components from 10^7 MHz to 10^8 MHz than the input signals shown in Fig. 4(b).

We can try different reduced orders for Ckt2 to set $q = 5$ and $q = 7$. The results are shown in Fig. 8 and Fig. 9. We can see that ETBR is still more accurate

than EKS as long as both of them use the same reduced order. And the CPU times of ETBR depends on the reduced order. If we want to achieve more accuracy, we need more reduced orders which results in more CPU times.

For the RLC circuits, ETBR also holds much more accuracy than EKS. Fig. 10 and Fig. 11 show the transient simulation waveforms and errors of both ETBR and EKS at the 200th node of Ckt9 (RLC).

There may be many high frequency components in the input signals in the real industry circuits. In this situation, we in general need more samplings to im-


Fig. 10 Transient waveform at the 200th node of Ckt9 (RLC).

Fig. 13 Input current waveform at node 17 of a real industry circuit.

Fig. 11 The simulation errors of ETBR and IEKS on Ckt9 (RLC).

Fig. 12 Transient waveform at node 17 of a real industry circuit.

prove the accuracy. Now we perform ETBR on a real industry circuit of 154514 nodes, 624 current sources and 25001 simulation time steps. We also perform latest UltraSim UPS (power network solver) on the same case. UltraSim UPS is a commercial power grid sim-

ulator from Cadence and the results of UltraSim UPS are considered as golden in this paper. The number of samplings in ETBR is still set to 10. But still the results are accurate enough as shown in Fig. 12. This circuit has rapid changing transient waveforms due to the reason that the current sources are changing very fast, as shown in Fig. 13.

Finally, we compare the CPU time of the two algorithms on a set of power grid networks up to one million nodes. The capacity of our implementation is mainly limited for Ckt1-Ckt8 by the physical memory of our machine (2GB).

Table 2 shows the CPU times of both ETBR (including the cost of FFT) and IEKS on the given set of circuits using the same reduction order $q = 6$. "—" means out-of-memory error. We find that EKS is a bit faster for small circuits. But for Ckt6 and larger circuits, the CPU times are almost the same for both methods. For the largest circuit Ckt8, EKS cannot even finish owing to the memory constraint; while ETBR runs through all the circuits. This clearly shows that ETBR is more memory efficient by using a non-LU decomposition solver than EKS.

Table 2 CPU times (in seconds) comparison of ETBR and IEKS ($q = 6$)

Test Ckts	ETBR (s)	EKS (s)
Ckt1	0.23	0.08
Ckt2	1.28	0.89
Ckt3	1.8	1.4
Ckt4	20.4	18.8
Ckt5	28.6	25.3
Ckt6	152	151
Ckt7	162	160
Ckt8	562	—
Ckt9 (RLC)	0.20	0.11
Ckt10 (RLC)	6.5	4.4

Table 3 shows the CPU times if parallelism is explored in ETBR. *PETBR* means parallelized ETBR. The results are collected on an Intel quad-core (3GHz

Table 3 CPU times (in seconds) comparison of ETBR, parallelized ETBR and IEKS ($q = 10$)

Test Ckts	# Nodes	# Sources	ETBR	PETBR	EKS
Ckt11	1,750,000	25,000	232	32	355
Ckt12	3,400,000	50,000	514	68	640
Ckt13	7,000,000	100,000	1349	167	—

CPU) workstation with 16GB memory. We assume that Step 3 in *Algorithm 1* can be fully parallelized. So the total CPU time of parallelized ETBR is the max CPU time out of all the sub-processes in parallelized Step 3 plus CPU time of serial parts in ETBR, such as FFT and SVD. For Krylov subspace method, such as EKS/IEKS, each moment is computed based on previous one, hence it is hardly to be parallelized. We can see that ETBR is now is a order of magnitude faster than EKS and ETBR.

4.2 Results on circuits with many different switching timings

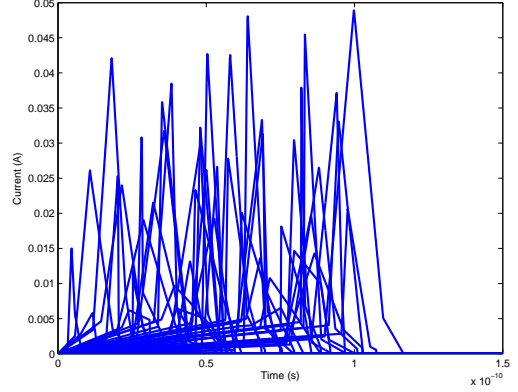
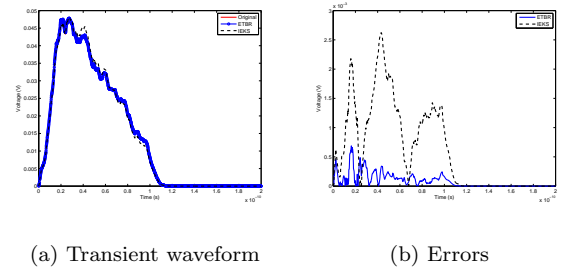
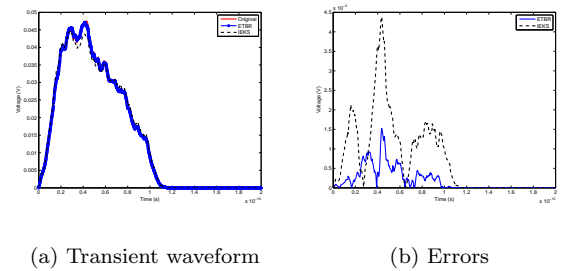
In this subsection, we show the results of ETBR and EKS are also very accurate for power grid circuits excited by input currents with many different switching timings (peaks).

The used benchmark circuit has 1000 nodes and each node has a current source, which switches at a different time (the peaks are different for each of them) as shown in Fig. 14. The resistor and capacitor values of this circuit are randomly generated. R is on the order of $10^{-2}\Omega$, and C is on the order of 10^{-15}F . The capacitance is really small. Fig. 15 and Fig. 16 show the simulation waveforms and errors on 100th and 300th nodes ($q = 5$). We can see ETBR is still very accurate.

We remark that if the circuit has very small capacitances such that the whole circuit become DC with respect to their input signal spectrum, then both ETBR and model order reduction in general cannot be applied in this case. But this is a very unrealistic case for general interconnect circuits modeled as RLC/RLCK circuits.

5. Conclusion

In this paper, we have proposed a new power grid analysis approach based on truncated balanced realization reduction techniques. The new simulation method, called *ETBR*, performs the reduction on the system before the transient simulation. But different from the existing extended Krylov subspace methods such as EKS or IEKS, it uses fast truncated balanced realization method on response Gramian to perform the reduction. As a result, ETBR can deliver more accurate results than the EKS method over large frequency range with similar computation costs. The ETBR is very amenable for threading-based parallel computing, as projection space vectors can be computed in parallel. This contrasts with all the Krylov subspace based methods like the EKS method, where moments have to

**Fig. 14** Transient waveforms of current sources switching at different time**Fig. 15** The transient waveform and errors at the 100th node ($q = 5$).**Fig. 16** The transient waveform and errors at the 300th node ($q = 5$).

be computed in a sequential order. Experimental results demonstrated that, given the same order of the reduced model, ETBR is more accurate than EKS especially for input sources rich in high-frequency components. If parallelism is explored, ETBR can deliver $10\times$ speedup over the EKS/IEKS method.

References

- [1] D. Li, S.X.D. Tan, and B. McGaughy, "ETBR: Extended truncated balanced realization method for on-chip power grid network analysis," Proc. Design, Automation and Test In Europe. (DATE), pp.432-437, 2008.

- [2] J.M. Wang and T.V. Nguyen, "Extended Krylov subspace method for reduced order analysis of linear circuit with multiple sources," Proc. Design Automation Conf. (DAC), pp.247–252, 2000.
- [3] Y. Lee, Y. Cao, T. Chen, J. Wang, and C. Chen, "HiPRIME: Hierarchical and passivity preserved interconnect macromodeling engine for RLKC power delivery," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol.24, no.6, pp.797–806, 2005.
- [4] "Umfpack." <http://www.cise.ufl.edu/research/sparse/umfpack/>.
- [5] S.R. Nassif and J.N. Kozhaya, "Fast power grid simulation," Proc. Design Automation Conf. (DAC), pp.156–161, 2000.
- [6] M. Zhao, R.V. Panda, S.S. Sapatnekar, T. Edwards, R. Chaudhry, and D. Blaauw, "Hierarchical analysis of power distribution networks," Proc. Design Automation Conf. (DAC), pp.150–155, 2000.
- [7] H. Li, Z. Qi, S.X.D. Tan, L. Wu, Y. Cai, and X. Hong, "Partitioning-based approach to fast on-chip decap budgeting and minimization," Proc. Design Automation Conf. (DAC), pp.170–175, June 2005.
- [8] J. Shi, Y. Cai, J. Fan, S.X.D. Tan, and X. Hong, "Pattern-based iterative method for extreme large power/ground analysis," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol.26, no.4, pp.680–692, Apr. 2007.
- [9] H.F. Qian, S.R. Nassif, and S.S. Sapatnekar, "Random walks in a supply network," Proc. Design Automation Conf. (DAC), pp.93–98, 2003.
- [10] W. Guo, S.X.D. Tan, Z. Luo, and X. Hong, "Partial random walk for large linear network analysis," Proc. IEEE Int. Symp. on Circuits and Systems (ISCAS), pp.V173–176, 2004.
- [11] L.T. Pillage, R.A. Rohrer, and C. Visweswariah, *Electronic Circuit and System Simulation Methods*, McGraw-Hill, New York, 1994.
- [12] S.X.D. Tan and L. He, *Advanced Model Order Reduction Techniques in VLSI Design*, Cambridge University Press, 2007.
- [13] P. Kongetira, K. Aingaran, and K. Olukotun, "Niagara: A 32-Way multithreaded sparcc processor," IEEE Micro, vol.25, no.2, pp.21–29, 2005.
- [14] B. Sinharoy, R. Kalla, J. Tendler, R. Eickemeyer, and J. Joyner, "Power5 system microarchitecture," IBM J. Research and Development, vol.49, no.4, pp.505–521, 2005.
- [15] B. Moore, "Principal component analysis in linear systems: Controllability, and observability, and model reduction," IEEE Trans. Automat. Contr., vol.26, no.1, pp.17–32, 1981.
- [16] J.R. Phillips and L.M. Silveira, "Poor man's TBR: a simple model reduction scheme," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol.24, no.1, pp.43–55, 2005.
- [17] J.R. Phillips and L.M. Silveira, "Poor man's TBR: a simple model reduction scheme," Proc. Design, Automation and Test In Europe. (DATE), pp.938–943, 2004.
- [18] J. Phillips, "Variational interconnect analysis via PMTBR," Proc. Int. Conf. on Computer Aided Design (ICCAD), pp.872–879, Nov. 2004.
- [19] Y. Saad, *Iterative methods for linear systems*, PWS publishing, 2000.