EMERGING TECH CONFERENCE – Edge Intelligence

Volume 03, 2024, pages 146 – 153

Proceedings of Emerging Tech Conference:
Edge Intelligence 2024

A low-rank balanced truncation approach for large-scale RLCk model order reduction based on extended Krylov subspace and a frequency-aware convergence criterion FinFETs

Christos Giamouzis, Dimitrios Garyfallou, Nestor Evmorfopoulos, and George Stamoulis
Dept. of Electrical and Computer Engineering, University of Thessaly, Volos, Greece
{cgiamouzis, digaryfa, nestevmo, georges}@e-ce.uth.gr

Abstract

Model order reduction (MOR) is essential in integrated circuit design, particularly when dealing with large-scale electromagnetic models extracted from complex designs. The nu merous passive elements introduced in these models pose significant challenges in the sim ulation process. MOR methods based on balanced truncation (BT) help address these challenges by producing compact reduced-order models (ROMs) that preserve the original model's input-output port behavior. In this work, we present an extended Krylov subspace based BT approach with a frequency-aware convergence criterion and efficient implemen tation techniques for reducing large-scale models. Experimental results indicate that our method generates accurate and compact ROMs while achieving up to ×22 smaller ROMs with similar accuracy compared to ANSYS RaptorX™ ROMs for large-scale benchmarks.

1 Introduction

Electromagnetic model extraction is crucial for designing and verifying integrated circuits (ICs), enabling precise simulation of the passive elements of the design. However, simulating extracted RLCk models with millions of elements and multiple ports is extremely computationally expen sive. Model order reduction (MOR) can reduce the complexity of such models while maintaining accurate input/output port behavior [1, 2]. By constructing reduced-order models (ROMs) that capture the essential dynamics of the original system, MOR can significantly reduce simulation time, enabling faster design iterations in IC development.

There are two main approaches to MOR. Moment matching (MM) methods are preferred for their efficiency, but they require manual selection of the number of moments [1]. Most importantly, they correlate the final ROM size with the number of moments and ports, limiting scalability. On the contrary, balanced truncation (BT) provides explicit theoretical bounds for the approximation error and is independent of the number of ports [2]. However, BT is restricted to small-scale models due to the high computational complexity of solving Lyapunov equations [2].

In this paper, we introduce an efficient low-rank BT technique to address the main scalability issue of the conventional BT approach. Specifically, we employ the extended Krylov subspace (EKS) method, which effectively solves the Lyapunov equations, drastically reducing the computational load of BT [3]. Additionally, we incorporate a frequency-aware convergence criterion, ensuring accuracy in the frequency

range of interest. Experimental evaluation indicates that the proposed method can be integrated into commercial extraction tools, such as the ANSYS RaptorX™ [4], to generate more compact ROMs of large-scale multi-port RLCk models.

2 Background

Consider the modified nodal analysis (MNA) description [5] of an n-node, m-branch (inductive), p-input, and q-output RLCk circuit in the time domain:

$$\begin{pmatrix} G_n & E \\ -E^T & 0 \end{pmatrix}\begin{pmatrix} v(t) \\ i(t) \end{pmatrix} + \begin{pmatrix} C_n & 0 \\ 0 & M \end{pmatrix}\begin{pmatrix} v(t) \\ i(t) \end{pmatrix} = \begin{pmatrix} B_1 \\ 0 \end{pmatrix} u(t),\ y(t) = \begin{pmatrix} L_1 & 0 \end{pmatrix}\begin{pmatrix} v(t) \\ i(t) \end{pmatrix} \quad (1)$$

where $\boldsymbol{G_n} \in R^{n\times n}$G (node conductance matrix), $\boldsymbol{C_n} \in R^{n\times n}$ (node capacitance matrix), $\mathrm{M} \in R^{\mathrm{m}\times\mathrm{m}}$ (branch inductance matrix), $\mathbf{E} \in R^{\mathrm{n}\times\mathrm{m}}$ (node-to-branch incidence matrix), $\mathbf{v} \in R^{\mathrm{m}}$ (vector of node voltages), i ∈ Rm (vector of inductive branch currents), $\mathbf{u} \in R^{p}$ (vector of v(t) ≡ dv(t) input excitations), $\boldsymbol{B_1} \in R^{n\times\mathrm{p}}$ (input-to-node connectivity matrix), $\mathbf{y} \in R^{q}$ (vector of output measurements), and $\boldsymbol{L_1} \in R^{\mathrm{q}\times\mathrm{n}}$ (node-to-output connectivity matrix). Moreover, we denote $\dot{}:\dot{v}(t) \equiv \frac{dv(t)}{dt}$ and $\dot{}:i(t) \equiv \frac{di(t)}{dt}$. If we now define the model order as N ≡ n+m, the state vector as $x(t) \equiv \begin{pmatrix} v(t) \\ i(t) \end{pmatrix}$, and also

$$G = -\begin{pmatrix} G_n & E \\ -E^T & 0 \end{pmatrix},\quad C \equiv \begin{pmatrix} C_n & 0 \\ 0 & M \end{pmatrix},\quad B \equiv \begin{pmatrix} B_1 \\ 0 \end{pmatrix},\quad \mathrm{L} \equiv \begin{pmatrix} L_1 & 0 \end{pmatrix})$$

then Eq. (1) can be written in the generalized state-space form, or so-called descriptor form:

$$C\frac{dx(t)}{dt} = Gx(t) + Bu(t),\quad y(t) = Lx(t) \quad (2)$$

The objective of MOR is to produce an equivalent ROM:

$$\tilde{C}\frac{d\tilde{x}(t)}{dt} = \tilde{G}\tilde{x}(t) + \tilde{B}u(t),\quad \tilde{y}(t) = \tilde{L}\tilde{x}(t) \quad (3)$$

where $\tilde{\mathbf{G}}, \tilde{\mathbf{C}} \in \mathrm{R}^{\mathrm{r}\times\mathrm{r}}$, $\tilde{\boldsymbol{B}} \in \mathrm{R}^{\mathrm{r}\times\mathrm{p}}$, $\tilde{\boldsymbol{L}} \in \mathrm{R}^{\mathrm{q}\times\mathrm{r}}$ the reduced order r << N, and the output error is bounded as:$||\tilde{\boldsymbol{y}}(t) - \boldsymbol{y}(t)||_2 < \varepsilon||\boldsymbol{u}(t)||_2$ for given **u**(t) and small ε. The output error bound in the frequency domain is:$||\tilde{y}(s) - \boldsymbol{y}(s)||_2 < \varepsilon||\boldsymbol{u}(s)"|_2$ via Plancherel's theorem [6]. If

$$\mathbf{H}(\mathrm{s}) = \mathbf{L}(\mathrm{s}\mathbf{C} - \mathrm{G})^{-1}\mathbf{B},\quad \tilde{\mathbf{H}}(\mathrm{s}) = \tilde{\mathrm{L}}\left(\mathrm{s}\tilde{\mathbf{C}} - \tilde{\mathbf{G}}\right)^{-1}\tilde{\mathbf{B}}$$

are the transfer functions of the original model and the ROM, the corresponding output error is:

$$||\tilde{\boldsymbol{y}}(s) - \boldsymbol{y}(s)||_2 = ||\tilde{\boldsymbol{H}}(s)\boldsymbol{u}(s) - \boldsymbol{H}(s)\boldsymbol{u}(s)||_2 \le ||\tilde{H}(s) - \boldsymbol{H}(s)||_\infty|\boldsymbol{u}(s)||_2$$

where $||\ .\ ||_\infty$ is the $\mathcal{L}_2$ matrix norm or H_∞ norm of a rational transfer function. Thus, to bound this error, we need to bound the distance between the transfer functions: $||\tilde{H}(s) - \boldsymbol{H}(s)||_\infty < \varepsilon < \varepsilon$.

3 MOR by Balanced Truncation

BT relies on the computation of the controllability Gramian **P** and observability Gramian **Q**, which are calculated as the solutions of the following **L**yapunov matrix equations [2]:

$$(\boldsymbol{C}^{-1}\mathbf{G})\mathbf{P} + \mathbf{P}(\mathbf{C}^{-1}\mathbf{G})^T = -(\boldsymbol{C}^{-1}\mathbf{B})(\boldsymbol{C}^{-1}\mathbf{B})^T,\ (\mathbf{C}^{-1}\mathbf{G})^T\mathbf{Q} + \mathbf{Q}(\boldsymbol{C}^{-1}\mathbf{G}) = -\boldsymbol{L}^{\mathrm{T}}\mathbf{L} \quad (4)$$

The controllability Gramian **P** describes the degree to which the states are controllable by the inputs, while the observability Gramian **Q** reflects the degree to which the states are observ able at the outputs. A ROM can theoretically be generated by eliminating the states that are difficult to control or observe. However, in the original state-space coordinates, certain states may be easy to control but difficult to observe, and vice versa. The process of "balancing" transforms the state vector to a new coordinate system, where the controllability and observ ability of each state are balanced, meaning each state is equally difficult to control and observe. An appropriate transformation **Tx**(t) exists, leading to the balanced state-space model:

$$\boldsymbol{TCT}^{-1}\frac{d(\boldsymbol{Tx}(t))}{dt} = \boldsymbol{TGT}^{-1}(\boldsymbol{Tx}(t)) + \boldsymbol{TBu}(t), \quad y(t) = \boldsymbol{L}T^{-1}(\boldsymbol{Tx}(t))$$

This balanced representation preserves the system's transfer function **H**(s) and simplifies to **P** = **Q**=*diag*($\sigma_1, \sigma_2, \ldots, \sigma_N$) [2], where σ_i are the Hankel singular values (HSVs). These HSVs are the square roots of the eigenvalues of the product **PQ**, i.e., $\sigma_i = \sqrt{\lambda_i(PQ)}$). In the above balanced model, the states with the largest HSVs are the easiest to both control and observe. If r of them are retained (truncating the $N - r$ states associated with the smallest HSVs), the error between the original and the reduced-order transfer functions is bounded as:

$$||\boldsymbol{H}(s) - \tilde{\boldsymbol{H}}(s)||_\infty \leq 2(\sigma_{r+1} + \sigma_{r+2} + \cdots + \sigma_N)$$

The above serves as an "a-priori" criterion that offers flexibility by allowing either the specification of a ROM size r to compute the error or a target error (target error) to determine the number r of HSVs to be preserved. This adaptability is a key advantage of BT over MM methods.

Algorithm 1 MOR by balanced truncation

Inputs: $\mathbf{G}, \mathbf{C}, \mathbf{B}, \mathbf{L}$
Outputs: $\tilde{\mathbf{G}}, \tilde{\mathbf{C}}, \tilde{\mathbf{B}}, \tilde{\mathbf{L}}$

1: Solve the Lyapunov equations to obtain the Gramian matrices $\mathbf{P}$ and $\mathbf{Q}$ [7]
2: Compute the SVD of the Gramian matrices: $\mathbf{P} = \mathbf{U}_P\mathbf{\Sigma}_P\mathbf{V}_P^T$ and $\mathbf{Q} = \mathbf{U}_Q\mathbf{\Sigma}_Q\mathbf{V}_Q^T$
3: Find the square root of the Gramian matrices: $\mathbf{Z}_P = \mathbf{U}_P\mathbf{\Sigma}_P^{1/2}$ and $\mathbf{Z}_Q = \mathbf{U}_Q\mathbf{\Sigma}_Q^{1/2}$
4: Compute the SVD of the product of the roots: $\mathbf{Z}_Q^T\mathbf{Z}_P = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$
5: Compute transformation matrices: $\mathbf{T}_{(r\times N)} = \mathbf{\Sigma}_{(r\times r)}^{-1/2}\mathbf{U}_{(r\times N)}\mathbf{Z}_Q^T$, $\mathbf{T}^{-1}_{(N\times r)} = \mathbf{Z}_P\mathbf{V}_{(N\times r)}\mathbf{\Sigma}^{-1/2}_{(r\times r)}$
6: Compute ROM: $\tilde{\mathbf{G}} = \mathbf{T}_{(r\times N)}\mathbf{G}\mathbf{T}^{-1}_{(N\times r)}$, $\tilde{\mathbf{C}} = \mathbf{T}_{(r\times N)}\mathbf{C}\mathbf{T}^{-1}_{(N\times r)}$, $\tilde{\mathbf{B}} = \mathbf{T}_{(r\times N)}\mathbf{B}$, $\tilde{\mathbf{L}} = \mathbf{L}\mathbf{T}^{-1}_{(N\times r)}$

The main steps of the BT procedure are summarized in Algorithm 1. The main limitation of BTisits high computational and memory cost, which makes it impractical for large-scale models (with N over a few thousand states). This is due to the computationally expensive operations required, such as solving Lyapunov equations and performing singular value decomposition (SVD), both of which have a complexity of O(N3). Additionally, they are applied on dense matrices, since the Gramians **P, Q** are dense even if the system matrices **C, G, B, L** are sparse.

However, the products $(\boldsymbol{C}^{-1}\mathbf{B})(\boldsymbol{C}^{-1}\mathbf{B})^T$ and $\boldsymbol{L}^T\boldsymbol{L}$ have a much lower numerical rank com pared to N, as $p,q << N$. This results in low-rank Gramians that can be approximated using low-rank techniques, significantly reducing the complexity and memory requirements for solving the Lyapunov equations and

performing SVD, which are now performed with a complexity of order k rather than N.

3.1. Low-rank BT MOR

The essence of low-rank BT MOR is to iteratively project the Lyapunov equations onto a lower dimensional Krylov subspace and solve the resulting small-scale equations to obtain low-rank approximate solutions of Eq. (4). The k-dimensional standard Krylov subspace is defined as:

$$\mathrm{K_k}(\mathbf{G_C}, \mathbf{B_C}) = \mathrm{span}\{\mathbf{B_C}, \mathrm{G_C}\mathbf{B_C}, \mathbf{G_C^2}\mathbf{B_C}, \dots, \mathbf{G_C^{k-1}}\mathbf{B_C}\}$$

where:$\boldsymbol{G_C \equiv C^{-1}G}$, $\boldsymbol{B_C \equiv C^{-1}B}$. If $\mathbf{K} \in R^{N\times k}$ (k << N) is a projection matrix whose columns span the *k*-dimensional standard Krylov subspace, then the projected Lyapunov equation (for the controllability Gramian **P**) onto $K_k(\boldsymbol{G_C}, \boldsymbol{B_C})$ is:

$$\left(\mathbf{K^T G_C K}\right)\mathbf{X} + \mathbf{X}\left(\mathbf{K^T G_C K}\right)^{\mathbf{T}} = -\left(\mathbf{K^T B_C B_C^T K}\right). \tag{5}$$

(the same holds true for the observability Gramian **Q** with $\boldsymbol{G_C^T}$, $\boldsymbol{L^T}$ in place of $\boldsymbol{G_C}$, B_C). The solution $\mathbf{X} \in \boldsymbol{R^{k\times k}}$ of Eq. (5) can be back-projected to the *N*-dimensional space to give an approximate solution $\mathbf{P} \approx \left(\boldsymbol{XX^T}\right)$P ≈ KXKT for the original large-scale Eq. (4), and a low-rank factor $\mathbf{Z} \in \boldsymbol{R^{N\times k}}$ of **P** can be obtained as $\mathbf{Z = KU\Sigma^{1/2}}$, where [**U,Σ,V**] = *SVD*(**X**) and $\mathbf{P \approx ZZ^T}$.

While the projection process is independent of the chosen subspace, its effectiveness heavily relies on it. The convergence to an accurate solution can be accelerated by enhancing the stan dard Krylov subspace $\boldsymbol{K_k(G_C, B_C)}$ with information from the subspace $K_k(G_C^{-1}, \boldsymbol{B_C})$, which corresponds to the inverse matrix G_C^{-1}, leading to the EKS [3, 8]:

$$\boldsymbol{K_k^C}(\mathbf{G_C}, \mathbf{B_C}) = \boldsymbol{span}\{\mathbf{B_C}, \mathbf{G_C^{-1}B_C}, \mathbf{G_C B_C}, \mathbf{G_C^{-2}B_C}, \mathbf{G_C^2 B_C}, \dots, \mathbf{G_C^{-(k-1)}B_C}, \mathbf{G_C^{k-1}B_C}\} \tag{6}$$

The EKS method (EKSM) begins with the vectors $\{\boldsymbol{B_C}, \boldsymbol{G_C^{-1}B_C}\}$. and iteratively builds an EKS $\boldsymbol{K_k^C(G_C, B_C)}$) of increasing dimension, solving the projected Lyapunov Eq. (5) in each iteration, until a sufficiently accurate approximation of the solution of Eq. (4) is achieved. The complete EKSM is presented in Algorithm 2. Below are some efficient implementation details:

- **Matrix inversion by linear solves**: Algorithm 2 uses the system matrices $\boldsymbol{G}, C, G^T$. $\boldsymbol{G^T}$instead of $\boldsymbol{G_C \equiv C^{-1}G}$ or $\boldsymbol{G_C^T \equiv (C^{-1}G)^T}$ since the (generally dense) inverse matrices are only required for products with *p* vectors (in step 2) and *2pj* vectors (in steps 4 and 11 of each iteration), which can be handled as linear solves like CY = R and **GY = R** (or $\mathbf{C^TY=R}$, $\mathbf{G^TY=R}$), using either direct or iterative methods [9].
- **Handling of sparse/dense matrices**: Matrix **M** of Eq. (1) is typically very dense due to the huge number of mutual inductances. To efficiently handle both sparse ($\mathbf{C_n}$) and dense (**M**) matrix blocks of **C**, we use specialized data structures and numerical techniques. This includes parallel CPU-optimized methods for sparse matrices and GPU-accelerated techniques [10] for dense matrices.
- **Solution of the small-scale Lyapunov equations**: To solve the small-scale (*2pj × 2pj*) Lyapunov equations in step 5 of each iteration, we employ the Bartels-Stewart algorithm [7].
- **Convergence criterion**: The error estimation [11] relies on the ROM transfer function ˜ H(s) and is

described by:

$$\max_{i=1,\dots,l} \frac{|\widetilde{\boldsymbol{H}_J}(s_i) - \widetilde{\boldsymbol{H}_{J-1}}(s_i)|_\infty}{|\widetilde{\boldsymbol{H}_J}(s_i)|_\infty}$$

where $\widetilde{\boldsymbol{H}_J}(s_i)$ is the ROM transfer function at the *j-th* iteration (calculated at frequency $s_i = 2\pi f_i$) and *l* is the number of evaluated frequency points evenly distributed across a frequency range [f_{min}, f_{max}]. The proposed criterion offers insight into the extent to which the transfer function changes between iterations at the frequencies of interest. Moreover, it proves to be practical and effective for circuit simulation problems, where designers are only interested in the circuit's behavior in certain frequency windows. The iterative procedure stops when the error remains below a certain threshold (*tol*) for three consecutive iterations.

4 Experimental Evaluation

We evaluated EKSM using RLCk models extracted from various circuits via ANSYS Rap torX™ [4]. The evaluated designs consist of a phase-locked loop (PLL), an analog mixer, a time-interleaved digital-to-analog converter (TI DAC), an injection-locked frequency multiplier (ILFM), a VGA circuit, hybrid couplers (HCs), Wilkinson power dividers (WPDs), and typical transceiver blocks, such as low-noise-amplifiers (LNAs) and oscillators (VCO). Their detailed characteristics are listed in Tables 1 and 2. Two experiments were conducted: in the first one, we used small-scale benchmarks (< 30K nodes), where the original and ROM transfer functions could be directly compared; in the second one, we used large-scale benchmarks and compared EKSM to golden RaptorX™ROMs through S-parameter plotting. For the reduction process, target error and tol were set to 1e-2 and the number of frequencies l was set to 20. Experiments were performed on a Linux server with a 2.80 GHz 16-thread CPU and 64 GB of memory.

For the first experiment, the results are presented in Table 1, where the error refers to the max relative error between the transfer functions of the original models and ROMs, which is

Algorithm 2 Extended Krylov subspace method for low-rank solution of Lyapunov equations

Input: $\mathbf{G}_C \equiv \mathbf{C}^{-1}\mathbf{G}, \mathbf{B}_C \equiv \mathbf{C}^{-1}\mathbf{B}$ (or $\mathbf{G}_C^T, \mathbf{L}^T$)
Output: $\mathbf{Z}$ such that $\mathbf{P} \approx \mathbf{Z}\mathbf{Z}^T$
1: $j = 1$; $p = size\_col(\mathbf{B}_C)$
2: $\mathbf{K}^{(j)} = Orth([\mathbf{B}_C, \mathbf{G}_C^{-1}\mathbf{B}_C])$
3: **while** $j < maxiter$ **do**
4: $\mathbf{A} = \mathbf{K}^{(j)T}\mathbf{G}_C\mathbf{K}^{(j)}$; $\mathbf{R} = \mathbf{K}^{(j)T}\mathbf{B}_C$
5: Solve $\mathbf{A}\mathbf{X} + \mathbf{X}\mathbf{A}^T = -\mathbf{R}\mathbf{R}^T$ for $\mathbf{X} \in \mathbb{R}^{2pj \times 2pj}$
6: **if** converged **then**
7: $[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] = \mathbf{SVD}(\mathbf{X})$; $\mathbf{Z} = \mathbf{K}^{(j)}\mathbf{U}\mathbf{\Sigma}^{1/2}$
8: **break**
9: **end if**
10: $k_1 = 2p(j-1)$; $k_2 = k_1 + p$; $k_3 = 2pj$
11: $\mathbf{K}_1 = [\mathbf{G}_C\mathbf{K}^{(j)}(:, k_1+1:k_2), \mathbf{G}_C^{-1}\mathbf{K}^{(j)}(:, k_2+1:k_3)]$
12: $\mathbf{K}_2 = Orth(\mathbf{K}_1)$ w.r.t. $\mathbf{K}^{(j)}$
13: $\mathbf{K}_3 = Orth(\mathbf{K}_2)$
14: $\mathbf{K}^{(j+1)} = [\mathbf{K}^{(j)}, \mathbf{K}_3]$
15: $j = j + 1$
16: **end while**

Table 1: Detailed characteristics of RLCk models and experimental results of EKSM

Model	Order	# ports	# mutual induct.	$\|\|\mathbf{G}_C\|\|_F$	$cond(\mathbf{G}_C)$	ROM order	Error	Reduction time	Memory
PLL @ 28 GHz	1474	4	251680	7.03e+21	4.41e+16	96	1.21e-03	3.8 s	72 MB
Mixer @ 28 GHz	1498	10	79794	4.13e+23	1.81e+18	100	2.04e-04	2.09 s	58 MB
TI_DAC @ 28 GHz	3869	160	365494	4.16e+22	9.62e+15	1280	1.14e-06	3 min	1.18 GB
LNA @ 56 GHz	4274	6	1988882	1.73e+24	1.37e+19	204	7.98e-04	31 s	341 MB
LNA @ 28 GHz	6956	6	5360490	1.55e+23	3.15e+18	144	4.48e-03	30 s	594 MB
ILFM @ 14 GHz	15665	11	18394794	4.68e+27	9.02e+22	176	1.19e-03	1 min	1.54 GB
LNA @ 2.4 GHz	25602	6	72959220	6.23e+23	5.14e+19	144	1.42e-03	3 min	6.62 GB

calculated as $||\boldsymbol{H}(s) - \widetilde{\boldsymbol{H}}(s)||_\infty / ||\boldsymbol{H}(s)||_\infty$ at the designated frequencies. For every benchmark, a base frequency of 100 MHz is chosen (f_{min} = 1e+8) and the maximum frequency is set to twice the resonance frequency of each circuit (e.g., f_{max} = 56e+9 for PLL@28GHz). As can be seen, EKSM generates accurate and compact ROMs across every type of benchmark with a maximum error below 0.14%. Additionally, the convergence criterion effectively strikes a balance between error and final ROM size while being computationally efficient. This is also visible through the performance results, where the reduction time remains below 3 min for benchmarks with less than 30K nodes and the memory requirements are not significantly high.

For the second experiment, the results are demonstrated in Table 2. The S-parameters plots of Figure 1 indicate that EKSM achieves accuracy close to that of RaptorX™ while producing on average ×13.2 more compact ROMs. Although EKSM has higher reduction time and memory requirements, they are still reasonable and can be significantly improved in futurk work.

Table 2: ROM order and MOR performance of EKSM vs RaptorX™

Model	Initial order	#ports	#mutual induct.	ROM order		Reduction time		Memory (GB)	
				RaptorX™	EKSM	RaptorX™	EKSM	RaptorX™	EKSM
VGA @ 28 GHz	95189	13	126766838	4744	286	1 min	11 min	32.63	19.14
HC @ 56 GHz	98024	5	165802476	1267	120	2 min	8 min	24.05	25.05
WPD @ 56 GHz	100888	4	193641938	765	40	3 min	4 min	24.79	29.76
VCO @ 13 GHz	104367	4	188436057	407	56	2 min	4 min	26.48	28.96
LNAC @ 56 GHz	128574	9	169339965	2172	378	1 min	19 min	25.82	26.01
WPD @ 28 GHz	129087	4	259462454	885	40	3 min	5 min	25.35	34.57
HC @ 28 GHz	134710	5	264162513	787	90	4 min	8 min	24.31	35.62
LNAC @ 28 GHz	162881	11	323090671	4768	308	6 min	27 min	78.52	49.58

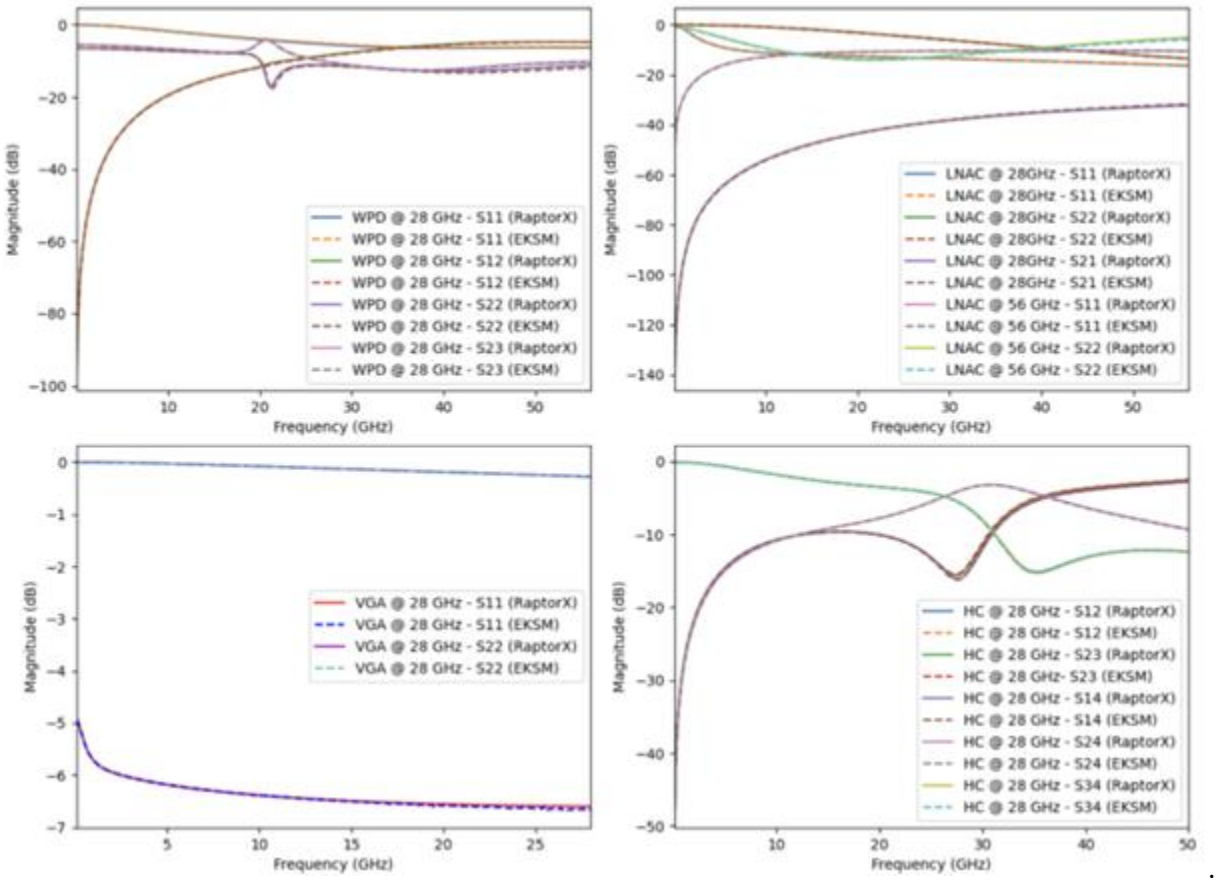

Figure 1: Comparison of accuracy between EKSM and RaptorX™ ROMs.

5 Conclusions

An alternative MOR technique for accurately reducing large-scale RLCk models is introduced. The proposed low-rank BT approach incorporates an iterative EKS projection method with a frequency-aware convergence criterion to produce accurate and compact ROMs. Experimental results demonstrate that our method provides up to ×22 smaller ROMs than ROMs obtained by ANSYS RaptorX™ for large-scale benchmarks with negligible deviations in S-parameters.

References

[1] A. Odabasioglu et al., "Prima: Passive reduced-order interconnect macromodeling algorithm," IEEE Trans. on CAD of Integrated Circuits and Systems, vol. 17, no. 8, pp. 645–654, 1998.

[2] S. Gugercin et al., "A survey of model reduction by balanced truncation and some new results," International Journal of Control, vol. 77, no. 8, pp. 748–766, 2004.

[3] C. Giamouzis et al., "Reduction of large-scale rlck models via low-rank balanced truncation," arXiv:2311.08478 [math.NA], 2023.

[4] "Ansys-RaptorX." [Online]. Available: www.ansys.com/products/semiconductors/ansys-raptorh

[5] C.-W. Ho et al., “The modified nodal approach to network analysis,” IEEE Transactions on Circuits and Systems, vol. 22, no. 6, pp. 504– 509, 1975.

[6] K. Gr¨ochenig, Foundations of Time-Frequency Analysis. Birkh¨auser, 2001.

[7] D. Lathauwer et al., “Computation of the canonical decomposition by means of a simultaneous generalized schur decomposition,” SIAM Journal on Matrix Analysis and Applications, vol. 26, no. 2, pp. 295–327, 2004. [

[8] C. Chatzigeorgiou et al., “Exploiting Extended Krylov Subspace for the Reduction of Regular and Singular Circuit Models,” in Proc. of the 26th Asia South Pacific Design Automation Conference, pp. 773–778, 2021.

[9] E. Bavier et al., “Amesos2 and Belos: Direct and Iterative Solvers for Large Sparse Linear Sys tems,” Sci. Program., vol. 20, no. 3, p. 241–255, 2012.

[10] D. Garyfallou et al., “A Combinatorial Multigrid Preconditioned Iterative Method for Large Scale Circuit Simulation on GPUs,” in Proc. of the 15th International Conference on Synthesis, Model ing, Analysis and Simulation Methods and Applications to Circuit Design, pp. 209–212, 2018.

[11] C. Giamouzis et al., “Low-rank balanced truncation of rlck models via frequency-aware rational krylov-based projection,” in Proc. of the 20th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design, 2024