Using the Superposition Property for Model Reduction of Linear Systems with a Large Number of Inputs*

***Peter Benner*†, *Lihong Feng*‡, *Evgenii B. Rudnyi*§**

At present, almost all model order reduction methods assume single-input single-output (SISO) systems or systems with a small number of inputs and outputs. Few methods can deal with systems with a large number of inputs and outputs. Multi-input multi-output (MIMO) systems appear for example in modeling of integrated circuits. The number of inputs and outputs sometimes is very large, even close to the number of state variables. In this paper we propose a novel model reduction technique which can efficiently perform model reduction for linear systems with a large number of inputs and outputs. Motivated by the superposition property of linear systems, model order reduction is performed separately with respect to each column of the input matrix. Then, the output response of the original multi-input system is approximated by the summation of the output responses of the reduced-order single-input systems corresponding to each column of the input matrix. The proposed method applies to both multi-input single-output systems

*This research is supported by the Alexander von Humboldt-Foundation and by the research network *SyReNe — System Reduction for Nanoscale IC Design* within the program *Mathematics for Innovations in Industry and Services* (Mathematik für Innovationen in Industrie und Dienstleistungen) funded by the German Federal Ministry of Education and Science (BMBF).

†Peter Benner and Lihong Feng are with Mathematics in Industry and Technology, Faculty of Mathematics, Chemnitz University of Technology, D-09107 Chemnitz, Germany; `benner@mathematik.tu-chemnitz.de`

‡Corresponding author, `lihong.feng@mathematik.tu-chemnitz.de`

§Evgenii B. Rudnyi is with CADFEM GmbH, Marktplatz 2, 85567 Grafing b. München, Germany; `erudnyi@cadfem.de`.

and multi-input multi-output systems. Theoretical analysis and numerical simulations show the advantages of the proposed method over the conventional model reduction method. A moment-matching property is proved to be satisfied by the reduced-order model.

1 Introduction

During the past 20 years, much progress has been made in the area of model order reduction for large-scale systems. Starting with [4, 5, 7, 8], model reduction for very large scale systems became feasible; nowadays, also large-scale nonlinear systems can be treated successfully, see, e.g., [6]. An even more recent development is the availability of methods for the more complicated parametric systems, e.g., [1]. All in all, the systems treatable by these methods have one common property, that is, they are either single-input single-output (SISO) systems or multi-input multi-output (MIMO) systems with a small number of inputs and outputs. If the numbers of inputs and outputs are very large, then many model reduction methods become inefficient.

Conventional moment-matching model reduction methods like PRIMA [8] compute a projection matrix whose columns form an orthonormal basis of a suitable Krylov subspace related to the moments of the system. The moments of the system are the coefficients in a power (Laurent) series expansion of the transfer function around some suitably chosen expansion points. Each of these coefficients in the series expansion is given as a matrix product with left-most factor equal to the input matrix and right-most factor equal to the output matrix and thus is a matrix of size $l \times r$, where l is the number of the inputs, r is the number of the outputs. If the number of independent columns in the input matrix is very large, then the number of columns in the projection matrix V increases very quickly, which may result in a model being not "reduced", the order of the model could be as large as that of the original system.

The recently proposed methods SVDMOR [2] and RecMOR [3] are designed to treat systems with large numbers of inputs and outputs. However, one common problem of both methods from [2] and [3] is that they may only be considered as heuristic as the computed reduced-order models do not preserve moment-matching properties which the conventional moment-matching method satisfies.

Regarding the drawbacks of the above mentioned methods, we propose an alternative model reduction method SPRIMA based on the superposition property, which is robust for linear systems with a large number of inputs and outputs. Furthermore, we prove that the reduced-order model derived by SPRIMA has the same moment-matching property as the conventional moment-matching method PRIMA has. Therefore, we not only provide a theoretical background of the method, but also expect a small local error of the reduced-order model. Finally, our method is suitable for parallel computing, which will further enhance the efficiency of the method.

In the following Section 2, we review the conventional moment-matching method PRIMA and analyze its problem for systems with large numbers of in-

puts and outputs. We propose the method SPRIMA based on the superposition property of linear systems in the third section, where also the moment-matching property of the method is stated. In Section 4, we provide a computational complexity analysis of SPRIMA and compare this with the complexity of PRIMA. We show with simulation results the advantages and robustness of our method in Section 5 and give conclusions in the end.

2 Conventional moment-matching model reduction method

A linear time-invariant (LTI) system is defined by

$$C\dot{\mathbf{x}}(t) + G\mathbf{x}(t) = Bu(t), \quad y(t) = L^T\mathbf{x}(t), \tag{1}$$

with constant matrices $C, G \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times l}$, and $L \in \mathbb{R}^{n\times r}$. Here, $\mathbf{x}(t) \in \mathbb{R}^n$ denotes the state vector of the system, while $u(t)$ and $y(t)$ stand for l inputs and r outputs, n is also called the order of the system.

In conventional moment-matching methods, like, e.g. PRIMA [8], the reduced-order model of the system is constructed using an approximation of the state vector $\mathbf{x} \approx V\mathbf{z}$:

$$\hat{C}\dot{\mathbf{z}}(t) + \hat{G}\mathbf{z}(t) = \hat{B}u(t), \quad \hat{y}(t) = \hat{L}^{\mathrm{T}}\mathbf{z}(t), \tag{2}$$

where $\hat{C} = V^TCV, \hat{G} = V^TGV \in \mathbb{R}^{q\times q}$, $\hat{B} = V^TB \in \mathbb{R}^{q\times l}$, and $\hat{L} = V^TL \in \mathbb{R}^{q\times r}$, $q \ll n$.

The basic steps are as follows. First, the transfer function of the above system is expanded in a series around an expansion point $s_0 \in \mathbb{C} \cup \{\infty\}$. Let $s = s_0 + \sigma$, then for small σ, we have

$$\begin{aligned} H(s_0+\sigma) = &\ L^{\mathrm{T}}[(s_0+\sigma)C+G]^{-1}B \\ = &\ L^{\mathrm{T}}[I+\sigma(s_0C+G)^{-1}C]^{-1}(s_0C+G)^{-1}B \\ = &\ L^{\mathrm{T}}[I-\sigma(s_0C+G)^{-1}C+\sigma^2[(s_0C+G)^{-1}C]^2-...](s_0C+G)^{-1}B \\ = &\ \sum_{i=0}^{\infty} \underbrace{L^{\mathrm{T}}[-(s_0C+G)^{-1}C]^i(s_0C+G)^{-1}B}_{:=m_i(s_0)}\,\sigma^i, \end{aligned}$$

where $m_i(s_0)$ for $i = 0, 1, 2, \ldots$ are called the moments of the transfer function about s_0.

The transformation matrix V for model order reduction is computed from the vectors which are associated with the moments, for example, using the expansion point $s_0 = 0$, we have

$$\mathrm{range}\{V\} = \mathrm{span}\{G^{-1}B, (G^{-1}C)G^{-1}B, \ldots, (G^{-1}C)^qG^{-1}B\}, \tag{3}$$

$q \ll n$. Once the matrix V is computed, the reduced-order model can be obtained by (2).

Moment-matching property

The goal of moment matching model reduction is the construction of a reduced-order system where certain moments $\hat{m}_i$ of the transfer function $\hat{H}$ of the reduced-order model match corresponding moments m_i of the original transfer function H: $\hat{m}_i = m_i,\ i = 0, 1, 2, ..., k$. It is known that by the above construction of V in (3), the reduced-order model matches $q+1$ moments, i.e. $\hat{m}_i(0) = m_i(0),\ i = 0, 1, 2, ..., q$ [8].

The problems of model reduction with the above moment-matching method for MIMO systems include: first, if the number of independent columns in the matrices B, L (i.e., the dimension of the subspace spanned by the columns of B, L) is of medium size, for example 50, and if model reduction requires higher order moments to get an accurate reduced-order model, then the size of the reduced-order model is probably not small (for example, if the size of the original system is $n = 5000$, the order of the reduced-order model matching 20 moments would be $q = 1000$). Moreover, the system matrices of the reduced-order model are usually dense, which makes the reduced-order system at least as expensive to simulate as the original system, contradicting the purpose of model reduction.

Second, if the number of independent columns in the matrices B, L are very large, for example 1000, then we cannot apply conventional model reduction at all, because we get a reduced-order model with almost the same size as the original system by just matching the first few moments. Therefore, we may derive a model which is not only unreduced, but also the sparsity of the original system is destroyed. An example in Section 5 shows this problem.

In the next section, we propose a model reduction method based on the superposition property of linear systems, which can be successfully applied to systems with a large number of inputs and outputs.

3 Model order reduction combined with superposition property

The superposition property is special to linear physical systems, see, e.g., [10]. It can be explained simply as follows: For a linear system (1), if $u_1(t) \rightarrow y_1(t)$ and $u_2(t) \rightarrow y_2(t)$ then $\alpha u_1(t) + \beta u_2(t) \rightarrow \alpha y_1(t) + \beta y_2(t)$, where $u_i, i = 1, 2$ are input signals, $y_i, i = 1, 2$ are corresponding output responses and α, β are two constant scalars.

Notice that the first equation of system (1) can be re-written as

$$C\dot{\mathbf{x}}(t) + G\mathbf{x}(t) = B\sum_{i=1}^{l} \tilde{u}^{(i)}(t),$$

where

$$\tilde{u}^{(i)}(t) = [0, \ldots, u_i(t), 0, \ldots, 0]^T,$$

and

$$u(t) = [u_1(t), u_2(t), \ldots, u_l(t)]^T.$$

The superposition property of linear physical systems tells us: the output response of the MIMO system (1) can be obtained by the sum of the output responses of the following single-input systems:

$$\begin{array}{rcl} C\dot{\mathbf{x}}(t) + G\mathbf{x}(t) &=& B\tilde{u}^{(i)} = b_i u_i(t), \\ y_i(t) &=& L^T\mathbf{x}(t), \end{array} \quad \text{for } i = 1, 2, \ldots, l, \tag{4}$$

i.e., $y(t) = y_1(t) + y_2(t) + \ldots + y_l(t)$, where b_i is the ith column of the input matrix B.

The transfer function of the original system can be obtained by the Laplace transform of $y(t)$ and its relation to the output response, that is,

$$\begin{array}{rcl} H(s)U(s) &=& Y(s) = Y_1(s) + \ldots + Y_l(s) \\ &=& H_1(s)U_1(s) + \ldots + H_l(s)U_l(s) \\ &=& [H_1(s), \ldots, H_l(s)] \cdot [U_1(s), \ldots, U_l(s)]^T, \end{array} \tag{5}$$

where $U(s) = [U_1(s), U_2(s), \ldots, U_l(s)]^T$ is the Laplace transform of the input $u(t)$ and $Y(s)$ is the Laplace transform of the output response of the original system. $Y_1(s), Y_2(s), \ldots, Y_l(s)$ are the Laplace transforms of the output responses of the single-input systems in (4), $H_1, H_2, \ldots, H_l$ are the corresponding transfer functions, and H is the transfer function of the original system. From the relations in (5), we see that

$$H(s) = [H_1(s), H_2(s), \ldots, H_l(s)]. \tag{6}$$

Next we propose to combine the superposition property with model reduction such that the original MIMO system (1) can be efficiently evaluated. If we separately apply a conventional model reduction method like PRIMA [8] to each single-input system in (4), then we can get approximate output responses $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_l$ of $y_1, y_2, \ldots, y_l$ from the reduced-order models of the single-input systems:

$$\begin{array}{rcl} \hat{C}\dot{\mathbf{z}}(t) + \hat{G}\mathbf{z}(t) &=& \hat{b}_i u(t), \\ \hat{y}_i(t) &=& \hat{L}^T\mathbf{z}(t), \end{array} \quad \text{for } i = 1, 2, \ldots, l. \tag{7}$$

The output response of system (1) is thus approximated by $y(t) \approx \hat{y}_1(t) + \hat{y}_2(t) + \ldots + \hat{y}_l(t)$. The transfer function of system (1) can also be approximately computed by the transfer functions of the reduced-order systems:

$$H(s) \approx \hat{H}(s) = [\hat{H}_1(s), \hat{H}_2(s), \ldots, \hat{H}_l(s)]. \tag{8}$$

We call the method proposed above SPRIMA, since it combines the **s**uperposition property with PRIMA. In the following, we prove the moment-matching property of SPRIMA. We show that SPRIMA preserves the moment-matching property of the conventional model reduction method PRIMA.

Theorem 1. *If the conventional moment-matching method PRIMA is applied to each single-input system in (4), and q_i moments of H_i are matched by the computed reduced-order transfer functions $\hat{H}_i(s)$ in (8), $i = 1, 2, \ldots, l$, then at least*

$$q := \min_{i=1,\ldots,l} q_i$$

moments of the original transfer function $H(s)$ (6) are matched by the reduced-order transfer function $\hat{H}(s)$ (8), i.e.

$$m_0(s_0) = \hat{m}_0(s_0), m_1(s_0) = \hat{m}_1(s_0), \ldots, m_{q-1}(s_0) = \hat{m}_{q-1}(s_0).$$

Proof. Recall that the transfer function of the original MIMO system has the following relation to the transfer functions of the single-input systems in (4):

$$H(s) = [H_1(s), H_2(s), \cdots, H_l(s)]. \tag{9}$$

SPRIMA obtains approximate transfer functions of $H_i(s),\ i = 1, 2, \cdots, l$, by model reduction of the single-input systems in (4). The approximate transfer function of (1) is

$$\hat{H}(s) = [\hat{H}_1(s), \hat{H}_2(s), \cdots, \hat{H}_l(s)]. \tag{10}$$

Theorem 1 can be easily proved by series expansions of $H_i(s)$ and $\hat{H}_i(s)$, respectively, $i = 1, 2, \ldots, l$, around the same expansion point and by using the moment-matching properties of PRIMA applied to H_i, $\hat{H}_i$. □

4 Computational complexity comparison

In this section, we compare the complexity of the proposed method SPRIMA with the conventional moment-matching method PRIMA. A rough analysis of the computational complexity of PRIMA shows:

I. When p moments are used to compute the projection matrix V as in (3), the Krylov subspace is composed of $l \cdot p$ vectors. The Arnoldi process is used in PRIMA to compute V, which requires $O(n^2 \cdot l \cdot p)$ flops to compute an orthogonal basis of the Krylov subspace: colspan$\{G^{-1}B, (G^{-1}G)G^{-1}B, \cdots, (G^{-1}C)^{(p-1)}G^{-1}B\}$. If G and C are sparse, then the complexity reduces to $O(n \cdot l \cdot p)$. (In both cases we use a (sparse) LU decomposition of G to compute G^{-1}. In order not to complicate the complexity analysis further, we assume the LU decomposition is computed prior to the Arnoldi process and is not included in the complexity analysis. If G is sparse, we assume that applying G or G^{-1} to a single vector requires $O(n)$ flops. This might be too rough an assumption in practice, but as this cost is the same for both approaches, it does not favor one approach over the other.)

II. If the $l \cdot p$ Krylov vectors are linearly independent, then the number of columns in the projection matrix V is $l \cdot p$. After the projection matrix V is computed, the computational complexity for deriving the reduced-order model is $O(n^2 \cdot l \cdot p) + O(n \cdot l \cdot p \cdot (l + r))$, where the first term reduces to $O(n \cdot l \cdot p)$ if G, C are sparse and we make the same assumptions as in I..

III. If the number of the columns in the projection matrix V is $l \cdot p$, then the order of the reduced-order model is equal to $l \cdot p$. Therefore, evaluating the reduced-order model in frequency domain analysis (or doing an implicit time step in

a time domain simulation) requires $O((l \cdot p)^3)$ flops if an LU decomposition is employed. (Note that $\hat{G}$ will in general be dense.)

In the above analysis, we assume that the input matrix B has full column rank. This is certainly justified in circuit analysis with a high number of terminals as the resulting inputs are usually independent of each other.

The complexity of SPRIMA is analyzed as follows. In order to attain the same accuracy (measured by the moment-matching property in Section 2) as PRIMA, the same number of p moments is used to generate the matrices $V_i,\ i = 1, 2, \ldots, l$ for each single-input system in (4).

a. There are l single-input systems in total, see (4). For each of them, a projection matrix V_i is computed using an orthonormal basis for the Krylov subspace colspan$\{G^{-1}b_i, (G^{-1}C)G^{-1}b_i, \cdots, (G^{-1}C)^{(p-1)}G^{-1}b_i\}$. Assuming again that the Arnoldi process is employed, the total complexity for computing all the projection matrices V_i is $l \cdot O(n^2 \cdot p)$ as in I. If G and C are sparse, the complexity reduces to $l \cdot O(n \cdot p)$. Note that the same (sparse) LU decomposition of G can be used in the Arnoldi processes for all individual single-input systems and thus needs to be computed only once in advance as in PRIMA.

b. If the p vectors $G^{-1}b_i, (G^{-1}C)G^{-1}b_i, \cdots, (G^{-1}C)^{(p-1)}G^{-1}b_i$ are linearly independent, the number of the columns in each matrix V_i is p. After V_i is computed, the computational complexity for deriving the l reduced-order models is: $l \cdot (O(n^2 \cdot p) + O(n \cdot p \cdot (r + l)))$, where as in II., the first term becomes $l \cdot O(n \cdot p)$ if G and C are sparse.

c. If the number of columns in each projection matrix V_i is p, then the order of each reduced-order model is p. Therefore the complexity for frequency or time domain simulation using the l reduced-order models is $l \cdot O(p^3)$ (per transfer function evaluation or time step).

From the above analysis, we see that the cost of obtaining the reduced-order models with either approach is similar (step I.,II. vs. step a.,b.). Therefore, the difference of computational complexity between PRIMA and SPRIMA reduces to the difference between step III. and step c., which is summarized in the following proposition.

Proposition 2. *If the number of the columns l in B is large, and the correlation between the columns is weak, then the computational complexity of evaluating the reduced-order model computed by PRIMA is $O(l^2)$ times higher than that of SPRIMA when matching the same number of moments.*

Let us consider an example: if there are $l = 150$ columns in B, and the number of independent columns in B is $l_0 = 100$, and if $p = 10$ moments are used to construct the projection matrix V, the order of the reduced-order system with PRIMA is $q = l_0 \cdot p = 1000$ while the order of each reduced-order single-input system in (7) is $q = p = 10$.

If one simulation step with the reduced-order model requires an LU decomposition, then following III., this comes at a cost of $O((l_0 p)^3) = O((1000)^3) = O(10^9)$ flops. Based on c., the complexity using the reduced-order model obtained with the superposition property is $l \cdot O(q^3) = 150 \cdot O(1000) = O(10^5)$. This shows that SPRIMA is much more efficient than PRIMA for systems with a large number of inputs.

5 Simulation results

In this section we show the efficiency of the proposed method SPRIMA with two circuit examples. The systems of both examples are in the form of the system in (1), with $L = B$. For the first example, the order of the original system is $n = 2952$, there are $l = 29$ columns in the input matrix B. The system of the second example has $l = 854$ columns in B, the order of the system is $n = 1726$. Both examples are related to industrial circuit models. The first one is provided by NEC Laboratories Europe in Sankt Augustin (Germany) while the second was produced with the circuit simulator TITAN developed by Qimonda AG, Neubiberg (Germany). Further details of the examples can not be provided due to non-disclosure agreements.

We give simulation results in Table 1 and Table 2, where $q^{(s)}$ is the order of the reduced-order models computed by SPRIMA, $q^{(d)}$ is the order of the reduced-order model obtained with PRIMA. *time* is the simulation time (in seconds) for solving the reduced-order models in the interested time intervals $t \in [0, 10^{-9}]$ in Table 1 and $t \in [0, 10^{-7}]$ in Table 2. (We only compare the time-domain simulation timings because we have analyzed before that the computational complexity for obtaining the reduced-order models is essentially equal for both approaches.) For the method SPRIMA, *time* is the total CPU time for simulating all the single-input reduced-order models. *error* is the error between the output response y_r of the reduced-order model and that of the original model y: $error = \|y - y_r\|_2 / \|y\|_2$. To be simple, we only show the output of the system corresponding to the first column of the output matrix L, the results for the other outputs are similar. The system of the first example is driven by ramp functions as inputs. The step functions are used as inputs in the second example. All simulations were performed on an IBM notebook with Intel CPU T2400, 1.83GHz, 1GB RAM.

The simulation results of the first example are listed in Table 1. The simulation time of SPRIMA changes very slowly with the increasing accuracy of the reduced-order system. Because the order of the reduced-order model increases fast with the increasing accuracy, PRIMA is nevertheless very sensitive to the accuracy of the reduced-order model. Note that when the accuracy of the reduced-order model is not high, PRIMA cost even less simulation time than SPRIMA, which seems to be not in accordance with the analysis in Section 4. This is due to the small size of the reduced-order system and the fact that in this example, the system has still relatively few inputs. However, when the accuracy increases, more moments need to be used to compute the transformation matrix V, the order of the reduced-order system by PRIMA increases quite fast, the corresponding simulation time of PRIMA thereby becomes larger than that of SPRIMA. The advantage of SPRIMA

Table 1. *Simulation results of the first example.*

Methods	$q^{(s)}$ / $q^{(d)}$	*error*	time (seconds)
SPRIMA	4	0.0032	0.17
SPRIMA	5	1.70×10^{-4}	0.19
SPRIMA	6	3.5×10^{-6}	0.19
SPRIMA	7	1.76×10^{-7}	0.25
SPRIMA	8	1.93×10^{-8}	0.22
SPRIMA	9	3.78×10^{-10}	0.23
SPRIMA	10	5.3×10^{-11}	0.25
PRIMA	58	0.0056	0.078
PRIMA	114	2.40×10^{-6}	0.22
PRIMA	139	5.17×10^{-9}	0.29
PRIMA	186	1.91×10^{-9}	0.57
PRIMA	232	1.17×10^{-9}	0.95

Table 2. *Simulation results of the second example.*

Methods	$q^{(s)}$ / $q^{(d)}$	*error*	time (seconds)
SPRIMA	10	0.45	54
SPRIMA	20	0.02	69.9
SPRIMA	30	3.4×10^{-4}	110
PRIMA	854	0.99	360.8
PRIMA	1519	4.2×10^{-4}	1899

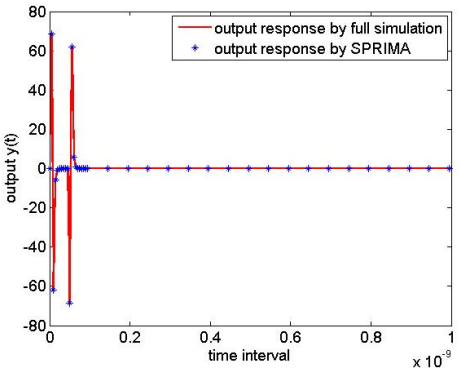

Figure 1. *Output response of the first example*

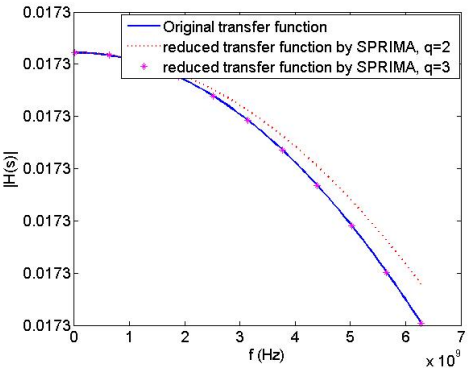

Figure 2. *Transfer function of the first example*

becomes obvious when the system has even more inputs, which can be seen from the second example in Table 2.

From the data in Table 2, the simulation time of PRIMA is much larger than that of SPRIMA. For the reduced-order models derived by PRIMA, if the error is required to be $O(10^{-4})$, then the order of the reduced-order model is 1519, which is a little less than the order $n = 1726$ of the original system. In this example, the system matrices $\hat{C}, \hat{G}$ of the reduced-order model are dense matrices as usual. However the original system matrices C, G are all sparse matrices. Therefore, as has been analyzed in Section 2, not only the original system is *unreduced*, but also the sparsity of the original system is destroyed. The system with dense matrices usually costs much more time to be simulated than the original system.

Figure 1 shows the output response of the first example. Being compared with the output response by simulating the original full system (1) (solid line), SPRIMA computes an accurate output response of the original system by much smaller reduced-order models. In contrast to the order of the original system $n = 2952$, the order of each reduced-order model is $q^{(s)} = 5$.

Figure 2 plots the transfer function of the first example, which corresponds to a particular single input port-single output port. The solid line is the transfer function computed from the original system. The other two lines are the transfer function computed by SPRIMA. The dotted line corresponds to the transfer function of a reduced-order model with order $q = 2$. The other line of star-markers is the transfer function of a reduced-order model with order $q = 3$. It catches the original transfer function very well, the relative error is $O(10^{-11})$.

6 Conclusions

We have proposed a simple, yet efficient model reduction method for linear systems with a large number of inputs and outputs. The method is based on the superposition property for linear systems. By using this property, the original MIMO system is decoupled into separate single-input systems. Model reduction is then

applied to each single-input system, the output response and the transfer function of the original system are approximated by the sum of the outputs or the combined transfer functions of the reduced-order single-input systems. The proposed model reduction method SPRIMA overcomes the difficulties of the conventional model reduction method applied to systems with a large number of inputs and outputs. The reduced-order model derived by SPRIMA satisfies the moment-matching property. It is easy to see from Section 3 that every single-input system in (4) can be dealt with independently, hence SPRIMA is suitable for parallel computing.

Future work will include efficient implementation of the SPRIMA method when the order of the original system is very large. In particular, when computing the separate Krylov subspaces for the single-input systems (4), only the starting vectors change so that efficient techniques from numerical linear algebra can be employed, like, e.g. recycling techniques [9].

Bibliography

[1] L. Daniel, O.C. Siong, L.S. Chay, K.H. Lee, and J. White. A multiparameter moment-matching model-reduction approach for generating geometrically parameterized interconnect performance models. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 22(5):678–693, 2004.

[2] P. Feldmann. Model order reduction techniques for linear systems with large numbers of terminals. *In Proc. Design Automation and Test in Europe*, 2004.

[3] P. Feldmann and Frank Liu. Sparse and efficient reduced order modeling of linear subcircuits with large number of terminals. *In Proc. International Conference on Computer-Aided Design*, 88–92, 2004.

[4] P. Feldmann and R.W. Freund. Efficient linear circuit analysis by Padé approximation via the Lanczos process. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 14:639–649, 1995.

[5] L.T. Pillage and R.A. Rohrer. Asymptotic waveform evaluation for timing analysis. *IEEE Trans. Comput.-Aided Design*, 9:325–366, 1990.

[6] M. Rewieński and J. White. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 22(2):155–170, 2003.

[7] K. Gallivan, E. Grimme, and P. Van Dooren. Asymptotic waveform evaluation via a Lanczos method. *Appl. Math. Lett.*, 7(5):75–80, 1994.

[8] A. Odabasioglu, M. Celik, and L.T. Pileggi. *PRIMA: passive reduced-order interconnect macromodeling algorithm. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 17(5):645-654, 1998.

[9] M.L. Parks, E. de Sturler, G. Mackey, D.D. Johnson, and S. Maiti. Recycling Krylov subspaces for sequences of linear systems. *SIAM J. Sci. Comput.*, 28(5):1651–1674, 2006.

[10] Oppenheim, A. V., and A. S. Willsky, with S. H. Nawab. Signals and Systems. 2nd ed. New Jersey: Prentice-Hall, 1997. ISBN: 0138147574.