TermMerg: An Efficient Terminal Reduction Method for Interconnect Circuits

Pu Liu, *Student Member, IEEE*, Sheldon X.-D. Tan, *Senior Member, IEEE*,
Bruce McGaughy, Lifeng Wu, Lei He *Member, IEEE*

***Abstract*— This paper proposes a novel method to efficiently reduce the terminal number of general linear interconnect circuits with a large number of input or output terminals considering delay uncertainty. Our new algorithm is motivated by the fact that terminal reduction can lead to more compact order reduced models and the observation that VLSI interconnect circuits have many similar terminals in terms of their timing and delay metrics due to their closeness in structure or due to mathematic discretization using meshing in finite difference or finite element scheme during the extraction process. The new method is based on the moments of the circuits as the metrics for the timing or delay. It then employs singular value decomposition (SVD) method to determine the best number of clusters based on the low-rank approximation. After this, the K-means clustering algorithm is used to cluster the moments of the terminals into the different clusters. The proposed method can work with any passive model order reduction and ensure the passive models. In contrast, we show SVDMOR does not generate passive models in general. Passivity enforcement in SVDMOR will significantly hamper the terminal reduction qualities. Experimental results on a number of real industry interconnect circuits demonstrate the effectiveness of the proposed method and show also that the proposed method is more accurate than SVDMOR when the used moment matrix does not give a good terminal correlations.**

***Index Terms*— Terminal Reduction, Singular Value Decomposition, K-Means**

I. Introduction

Complexity reduction and compact modeling of interconnect networks have been an intensive research area in the past decade due to increasing signal integrity effects and rising electro and magnetic couplings modeled by parasitic capacitors and inductors. Most of previous research works mainly focus on the reduction of the internal circuitry by various reduction techniques. The most popular one is based on subspace projection [3], [5], [9], [12], [14]. Projection-based method was pioneered by Asymptotic Waveform Evaluation (AWE) algorithm [12] where explicit moment matching was used to compute dominant poles at low frequency. Later on more numerical stable techniques are proposed [3], [5], [9], [14] by using implicit moment matching and congruence transformation.

Some preliminary results of this paper will appear in *Proc. Int. Conf. Computer Aided Design (ICCAD'05)* [8]

Pu Liu, Sheldon X.-D. Tan, are with Department of Electrical Engineering, University of California, Riverside, CA 92521 USA (e-mail: {uliu,stan}@ee.ucr.edu)

Lifeng Wu and Bruce McGaughy are with Cadence Design Systems Inc.

Lei He are with Department of Electrical Engineering, University of California, Los Angeles, CA 90095, USA (e-mail: {lhe}@ee.ucla.edu)

This work is funded by NSF CAREER Award CCF-0448534, UC Micro #05-111 via Cadence Design System Inc.

However, nearly all existing model order reduction techniques are restricted to suppress the internal nodes of a circuit. Terminal reduction, however, is less investigated for compact modeling of interconnect circuits. Terminal reduction is to reduce the number of terminals of a given circuit under assumption that some terminals are similar in terms of performance metrics like timing or delays. Such reduction will lead to some accuracy loss. But terminal reduction can lead to more compact models after traditional model order reduction is applied to the terminal-reduced circuit as shown in Fig. 1.

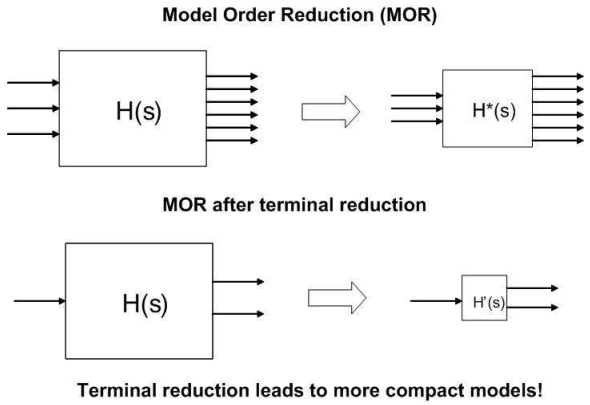

Fig. 1. Terminal reduction versus traditional model order reduction.

For instance, if we use subspace projection methods like PRIMA [9] for the model order reduction, less terminal count will lead to smaller reduced models given the same order of block moment requirement for both circuits. The reason is that for every block moment order increase, PRIMA will generate m new poles in the reduced models where m is the number of terminals. Hence less terminals will directly lead to less poles used in the reduced models.

For terminal reduction, another question one may have is that: are there many similar terminals in many practical interconnect circuits? One important observation we have is that many terminals in practical interconnect circuits are close to each other structurally or they are extracted by using mathematic discretization by volume or surface meshing in methods like finite difference and finite element scheme. As a result, their electrical characteristics are also similar in a well-designed VLSI system. For instance clock sinks in clock networks, substrate plane, critical interconnects in the memory circuits like word or bit lines are among those interconnects. Recent studies [2], [4] show that there exists a large degree of correlation between the various input and output terminals.

So a low-rank approximation was performed on the input and output position matrices before the model order reduction process. However, the low-rank approximation in the existing methods are only based on the DC or a specific order of moments of responses. So the port-reduced systems may not correlate well with the original systems in terms of timing or delay.

In this paper we present a novel terminal reduction method called *TermMerg*. The new terminal reduction method is based on the observation that if we allow some delay tolerance or variations, which actually can't be avoided in today's VLSI chip manufacture and working environments, some of the terminals with similar timing responses actually can be suppressed or merged into one representative terminal during the reduction without affecting their modeling functionality. In contrast to the existing terminal reduction methods, the new approach uses high order moments as timing and delay metrics. Specifically, given some delay tolerances or variations, *TermMerg* employs singular value decomposition (SVD) method to determine the number of clusters based on the low-rank approximation. Then the K-means clustering algorithm is used to clusters the moments of the terminals into different clusters. After the clustering, we pick one terminal that could best represent other terminals for each cluster.

We notice that a similar approach was applied to video image processes where SVD is followed by a K-mean clustering algorithm [7] [1]

The paper is organized as follows. In Section III, we first briefly review the concept of moments and then we present how the input and output moment matrices are constructed for terminal reduction. Section IV presents a SVD-based method to obtain the optimal number of clusters for the proposed terminal merging algorithm. In section V, we describe the K-means based *TermMerg* algorithm to find the representative terminals for each cluster. In section VI, we present the whole terminal reduction flow and then discuss some practical issues associated with the implementation. In section VII we analyze the passivity issues for both algorithms. The experimental results and conclusions are presented in section VIII and section IX, respectively.

II. Review of the SVDMOR method

In this section, we briefly review the SVDMOR method for terminal reduction, which was proposed recently for reducing the terminals of interconnect circuits [2], [4].

For a linear RLC interconnect network with p input and q output terminals, we can apply Modified Nodal Analysis to formulate it into the state space equation form

$$\begin{array}{l} G\mathbf{x}(t) + C\dot{\mathbf{x}}(t) = B\mathbf{u}(t) \\ \mathbf{y}(t) = L\mathbf{x}(t), \end{array} \tag{1}$$

where $G \in R^{n\times n}$ and $C \in R^{n\times n}$ are the conductive and storage element matrices. $L \in R^{q\times n}$ and $B \in R^{n\times p}$ are the output and input position matrices. $\mathbf{y}(t) \in R^q$, $\mathbf{u}(t) \in R^p$. State variables $\mathbf{x}(t) \in R^n$ can be nodal voltages or branch currents of the linear circuit.

[1]Our work was done without be aware of this work.

The circuit transfer function is

$$H(s) = L(G + Cs)^{-1}B. \tag{2}$$

Then the ith block moment of the system is defined as

$$\mathbf{m}_i = L(-G^{-1}C)^i G^{-1}B, \tag{3}$$

which is a $q \times p$ matrix function.

The block moment $\mathbf{m}_i$ can be directly computed in a recursive way

$$\begin{array}{l} \mathbf{x}_0 = G^{-1}B;\ \mathbf{m}_0 = L\mathbf{x}_0 \\ \mathbf{x}_1 = -G^{-1}C\mathbf{x}_0;\ \mathbf{m}_1 = L\mathbf{x}_1 \\ \cdots \\ \mathbf{x}_i = -G^{-1}C\mathbf{x}_{i-1};\ \mathbf{m}_i = L\mathbf{x}_i \text{ for } i > 0, \end{array} \tag{4}$$

SVDMOR basically exploits the fact that many terminals are not independent in terms of their timing information, which can be reflected in their frequency domain moments. As a result, we can perform the singular value decomposition (SVD) on a block moment of specific order. For instance, if we perform the SVD on the 0th order block moment (DC response) $\mathbf{m}_0$, we have

$$\mathbf{m}_0 = LG^{-1}B = U\Sigma V^T, \tag{5}$$

where U and V are orthogonal matrices and Σ is a diagonal matrix with singular values in the diagonal in a decreasing order. If there are k dominant singular values and we can use a k-rank matrix (a $k \times k$ full rank matrix) to approximate the original $\mathbf{m}_0$ based on the SVD theory as

$$\mathbf{m}_0 = U\Sigma V^T \approx U_k\Sigma_k V_k^T. \tag{6}$$

Notice that U_k is $q \times k$ matrix and V_k^T is a $k \times p$ matrix and Σ_k is a $k \times k$ matrix. After this, we can have the following expressions

$$B = B_b V_k^T, \tag{7}$$

$$L^T = L_c U_k^T, \tag{8}$$

where $B_b \in R^{n\times k}$ and $L_c \in R^{n\times k}$ are obtained using the Moore-Penrose pseudoinverse of V_k.

$$B_b = BV_k(V_k^T V_k)^{-1}, \tag{9}$$

$$L_c = L^T U_k(U_k^T U_k)^{-1}. \tag{10}$$

The circuit transfer function now becomes

$$H(s) = U_k L_c^T (G + Cs)^{-1} B_b V_k^T. \tag{11}$$

Notice that the transfer function $H_r(s)$,

$$H_r(s) = L_c^T (G + Cs)^{-1} B_b, \tag{12}$$

which is inside (11), is a $k \times k$ matrix transfer function, which actually is the terminal-reduced transfer function of (2) and can be reduced by traditional Krylov subspace based model order reduction methods. If the reduced transfer function of (12) is $\hat{H}_r(s)$, then the final order reduced transfer function is

$$\hat{H}(s) = U_k \hat{H}_r(s) V_k^T. \tag{13}$$

SVDMOR method has several limitations. First, when the input and output terminals are quite different (especially in terms of numbers), SVDMOR does not work very well as

SVDMOR performs the terminal reduction on both input and output responses at the same time. As a result, it can only approximate well for one type of terminals which have smaller terminal counts (as the ranke of $\mathbf{m_i}$ typically is determined by the smaller dimentions of its rows or columns). We will compare SVDMOR with the new method in Section VIII-A.

The second problem with SVDMOR is that it is difficult to enforce the passivity during the combined terminal and model order reduction and passivity terminal reduction of SVDMOR leads to less effect terminal reduction. We will discuss the passivity issues in Section VII.

III. Input and Output Moment Matrices

Our task is to find the terminals with similar delay or timing behaviors such that they can be viewed as one terminal if some delay uncertainty are allowed. We focus on the timing metric of terminals and look at their timing responses due to step or impulse inputs. But our method can be applied to other metrics of interest.

Ideally, the delay or timing information should be represented by waveforms in the time domain . But this does not give us the best representation for our terminal merging method because all the waveforms have to be computed first. It is also difficult to compare two transient waveforms in general. Instead, we use terminal response moments in frequency domain to represent their time-domain response information.

Our algorithm is based on the observation that if two terminals have similar timing or delay, then they should have the similar moments (vectors) numerically. Remember that the moments computed in (17) represent the impulse responses of outputs due to inputs. It is well know that the 1^{th} moment m_1 represents the first-order delay approximation, or the Elmore delay, of the corresponding output with respect to a specific input. Higher order moments represent more detailed timing and delay information. So moment vector is an ideal expression of timing information for our terminal merging algorithm.

Since we need to merge both input and output terminals, we need to present the timing information for both input and output terminal. As a result, we have two moment matrices, the input moment matrix M_I and output moment matrix M_O.

For a 1×1 system, the system's transfer function $H(s)$ could be expanded into a Taylor series around $s = 0$. The coefficient of s^i in the series expansion is the i^{th} moment of the transfer function:

$$H(s) \quad = m_0 + m_1 s + m_2 s^2 + m_3 s^3 + \dots \tag{14}$$

For a general linear (linearized) time-invariant network with p input and q output terminals, we can describe it as follows.

$$\begin{aligned} \dot{\mathbf{x}} &= A\mathbf{x} + B\mathbf{u} \\ \mathbf{y} &= C\mathbf{x} \end{aligned} \tag{15}$$

Where $\mathbf{x}$ is a n-dimensional state vector, $\mathbf{u}$ is a p-dimensional input vector, and $\mathbf{y}$ is a q-dimensional output vector. The transfer function by Laplace transformation is

$$H(s) = C(sI - A)^{-1}B \tag{16}$$

If we expand the above equation in a Taylor's series at $s = 0$, we get the moments at various terminals:

$$\begin{aligned} &\mathbf{m}_0 = -CA^{-1}B \\ &\mathbf{m}_1 = -CA^{-2}B \\ &\vdots \\ &\mathbf{m}_{r-1} = -CA^{-r}B \\ &\vdots \end{aligned} \tag{17}$$

Each moment $\mathbf{m}_i$ is a $q \times p$ matrix,

$$\mathbf{m}_i = \begin{bmatrix} m^i_{1,1} & m^i_{1,2} & \dots & m^i_{1,p} \\ m^i_{2,1} & m^i_{2,2} & \dots & m^i_{2,p} \\ \vdots & \vdots & \vdots & \vdots \\ m^i_{q,1} & m^i_{q,2} & \dots & m^i_{q,p} \end{bmatrix} \tag{18}$$

where each column j in $\mathbf{m}_i$ represents the moment vector of all output terminals due to the input terminal j and each row k in $\mathbf{m}_i$ represents the moment vector at the output terminal k due to all input terminals. Then a moment matrix can be written as

$$M = \begin{bmatrix} \mathbf{m}_0 & \mathbf{m}_1 & \dots & \mathbf{m}_{r-1} & \dots \end{bmatrix} \tag{19}$$

In order to perform terminal reduction for both inputs and outputs, different moment matrices are constructed. For the output terminal reduction, we define the *output moment matrix* M_O as:

$$M_O = \begin{bmatrix} \mathbf{m}_0^T \\ \mathbf{m}_1^T \\ \vdots \\ \mathbf{m}_{r-1}^T \end{bmatrix} \tag{20}$$

where each column j represents a moment series of output node j due to all input's stimuli. Notice that in this way, the output terminal's responses are with respect to all the inputs to make sure they are similar under all the inputs.

Similarly, for input terminal reduction, the *input moment matrix* M_I is defined as:

$$M_I = \begin{bmatrix} \mathbf{m}_0 \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_{r-1} \end{bmatrix} \tag{21}$$

where each column k represents a moment series at all output's nodes due to an input node k.

To determine the best order of moments: r, we use the following rule: the number of moments from all the inputs should be equal or large than the number of terminals to be merged. The reason is that in the worst case where no terminals can be merged, we should be able to distinguish all the terminals using the moment information. This will become more clear in the following section. As a result, we have

$$rp \geq q \text{ for } M_O \tag{22}$$

$$p \leq rq \text{ for } M_I \tag{23}$$

When $p = q$, we can simply set $r = 1$ to satisfy (22) and (23). If we have very large input and output terminals, then we can't apply SVD to do the terminal reduction directly (this is

also true for [2]) as SVD has about $O(n^3)$ complexity, where n is the size of the matrix [6]. However, we can do the terminal reduction in a hierarchical way: we can partition the circuit into k subcircuits such that the terminal in each subcircuit is small enough for SVD as shown in [4]. More discussions on other special cases will be presented in Subsection VI-C.

Moments can be efficiently computed by recursively solving the given circuits using traditional SPICE-like simulation technique [12]. After we obtain the moment matrix as shown in (20) or (21), we proceed to find the optimum number of clusters using singular value decomposition method to be shown in the next section.

IV. Determination of Cluster Number By SVD

In this section, we present how to find the best number of independent clusters by using singular value decomposition on the input and output moment matrices discussed in the previous section.

If two terminals have similar timing responses, it means that their moments have very similar values. If we have a number of terminals with similar timing behaviors, their moment matrix, where each moment series is a column or a row, will be a low-rank matrix. Singular value decomposition is very efficient to deal with rank-deficient matrices and it can reveal a great deal about the ranks and structure of a matrix, which motivate us to find the optimal number of clusters based on the moment matrices.

For a $m \times n$ matrix A, the SVD decomposition of A is

$$A = U_{m\times m}\Sigma V^T_{n\times n} \tag{24}$$

where $U_{m\times m}$ and $V_{n\times n}$ are orthogonal matrices, $U^T_{m\times m}U_{m\times m} = I$ and $V^T_{n\times n}V_{n\times n} = I$. $\Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_{min(m,n)})$, σ_i is called singular values and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{min(m,n)}$.

Before we proceed to present our cluster number determination method, we review the important result from the SVD decomposition [6].

For a SVD decomposition of a matrix A, if $k < r = rank(A)$ and

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T, \tag{25}$$

then

$$\min_{rank(B)=k} \| A - B \|_2 = \| A - A_k \|_2 = \sigma_{k+1} \tag{26}$$

where $\{u_i\}$ and $\{v_i\}$ are the lift and right singular vectors respectively. (26) basically reflects the fact that the rank-k approximation matrix A_k, is just σ_{k+1} away from original matrix A in terms of norm-2 distance.

In summary, for a matrix, A, SVD can basically do two things. The first thing is that SVD can tell us how many row or column are independent (the true rank of A) in a numerical way. Second, SVD can give a good low-rank approximation to the original matrix $A \approx A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$. But it is difficult to find the direct relationship between those left and right singular vectors u_i and v_i with the columns of the matrix A as those vectors are dominant (independent) columns of the original matrix in a different coordinate. But the rank information is still valid for the columns in the original matrix A.

In our problem, basically we only use the true (low) rank information provided by the SVD instead of the singular vectors. Specifically, we look at the singular values and select the sufficient small singular value σ_{k+1} to select the correct rank information. In the moment matrix, each column represents moment response for a input or a output terminal. If two terminals have the same response, they should have the same column numerically and thus they are dependent. So the selected rank number k essentially can be viewed as the number of clusters we expect as SVD essentially reveals the true rank of a matrix in a numerical way.

Notice in general the rank of a set of vectors (column of a matrix) is different from the similarity defined by the Euclidean distance of the vectors used in the K-mean method. For instance, for n 1x1 column vectors (each vector is just a scalar), their rank is 1 while each of the elements in the each vector may be different. However, as we increase the dimensions of the vectors, the fact that two vectors become correlated or dependent will become more close to their similarity when we compare more elements for any two vectors. This can also be easily understood by the fact that with larger vector space dimensions, two vector will be easier to become independent. This is especially true for moment vectors where each element in a vector is a specific order of moments with physical meaning. For instance, the first order moment is Elmore delay. As we use higher order moments in our approach, it is very unlikely that two moments are correlated but they are not similar. As a result, the rank information from SVD on the input and output matrices with high order moments will give us a good approximation on the number of truly unique moment vectors we may have, which is consistent with the follow-up similarity-based K-mean method.

We notice that a similar approach was applied to efficient clustering of video images to generate a compact representation of a video sequence to enable fast access to video contents [7]. The method performs the SVD on a transformed matrix, which has the one-to-one corresponding to the original matrix for its all the column vectors and then performs the K-mean clustering on the transformed matrix to speed up the clustering process. But the clustering essentially is still performed on the columns of the original matrix.

We set a threshold ζ to denote the small singular value σ_{k+1}. Typically ζ is set to 10^{-3} in our experiments, which means we regard σ_{k+1} as the sufficient small singular value only if $\sigma_{k+1} \leq \zeta$.

In our method, we select the approximation order not only based on the absolute value of the sufficient small singular value, but also on the relative ratio between two adjacent singular values. If the relative ratio between two adjacent singular values is close to 1, that means there is no big difference if the approximation order is increased by one. In this condition, we will keep increasing the approximation order until the ratio becomes small enough. Thus we first define a threshold ε. Then we compare two adjacent singular values σ_{k+1} and σ_k. If $\sigma_{k+1}/\sigma_k \leq \varepsilon$ and $\sigma_{k+1} \leq \zeta$, then the k is our choice. Typically ε is set to 10^{-3} in our experiments.

V. K-MEANS BASED CLUSTERING ALGORITHM

After we determine the number of clusters for the output(or input) terminals, say k, we proceed to group the output(or input) terminals into the k clusters.

In our approach, we apply a so-called K-means algorithm to determine each cluster [1]. K-means is the widely used clustering algorithm for partitioning (or clustering) N data points into k disjoint S_j subsets containing N_j data points to minimize the sum-of-squares criterion:

$$J = \sum_{j=1}^{k} \sum_{i \in S_j} |\mathbf{x}_i - \varphi_j|^2 \tag{27}$$

where $\mathbf{x}_i$ is a vector representing the i^{th} data point, and φ_j is a vector, representing the geometric centroid of the data points in S_j. It has wide applications in data mining and data analysis.

However, the K-means clustering algorithm is very sensitive to the initial choice of cluster centers and the number of clusters [1]. Only the appropriate number of clusters produces the best approximation result. Fortunately in our method, cluster number k can be first obtained from SVD decomposition on the moment matrix as mentioned above.

For a general moment matrix M ($M = M_O, or M = M_I$ for different terminal reductions), it has l terminals to be merged. Let $M = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_l]$.

The proposed clustering algorithm is shown in Fig. 2 based on K-means clustering scheme.

K-MEANSCLUSTER
CONSTRUCT_CLUSTER(k) 1 select k seed vectors out of $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_l$ as k centroids 2 put rest of unselected vectors into nearest cluster S_j 3 compute φ_j for each cluster S_j 4 **do** re-cluster all $\mathbf{c}_i$ into k clusters according to φ_j of each cluster 5 recompute φ_j for each cluster S_j 6 **until** no change in φ_j 7 **return** $S_1, S_2, \ldots, S_k$ and $\varphi_1, \varphi_2, \ldots, \varphi_k$
SELECT_REP_VECTOR($S_1, S_2, \ldots, S_k$) 1 **for** $i \in S_j$, compute $\|d_i\|^2 = \|\mathbf{c}_i - \varphi_j\|^2$ 2 find $R_j = \mathbf{c}_i$ so that $\min\{\|d_i\|^2\}$ 3 **return** R 4 end

Fig. 2. K-means based clustering algorithm.

There are two steps (algorithms) in our clustering method. The first step CONSTRUCT_CLUSTER(k) clusters the given l vectors into k clusters ($l > k$). The second step SELECT_REP_VECTOR($S_1, S_2, \ldots, S_k$)) takes the output of the clustering algorithm as the input and finds the representative vector for each cluster. The representative vectors R_j will be kept during the terminal reduction. All the other unselected vector will be suppressed.

The basic idea of CONSTRUCT_CLUSTER(), is to dynamically find the k clusters so that all the vectors in a cluster has the closest distance to its geometric centroid vector φ_j of cluster S_j. CONSTRUCT_CLUSTER() uses an iterative algorithm that minimizes the sum of distances from each vector to its cluster centroid vector φ_j, over all clusters. This algorithm moves vectors among clusters until the sum cannot be decreased any more.

The second algorithm SELECT_REP_VECTOR () finds the representative vector for each given cluster S_j. This has been achieved by calculating the distance between the centroid and each vector belonging to this cluster to find the terminal with the closest distance to the centroid.

Since we use the representative terminal to represent all the other terminals in a cluster, the transfer functions of the reduced one (thus their zero) will be different than their original ones. But we ascertain that their expanded moment form will be as close as possible among terminals in a cluster (as we work on the moment matrix). Hence we expect the zeros of the two transfer functions (for the representative one and the reduced one) will be similar.

VI. TERMMERG ALGORITHM

In this section, we first present the whole terminal reduction flow of the TERMMERG algorithm. Then we give the compact modeling flow, which combines the terminal reduction with traditional model order reduction. Finally we discuss some practical considerations of the algorithm.

A. TermMerg Algorithm Flow

In this subsection, we present the flow of the TermMerg method.

TERMMERG ALGORITHM

1. Construct the input and/or the output moment matrices.
2. Perform the SVD on the input and/or the output moment matrices to find the best cluster numbers.
3. Invoke K-MEANSCLUSTER to find the all the representative terminals for each cluster.

After the terminal reduction, the input position matrix B and the output position matrix C in (15) will be modified to include only the representative terminals. We can define terminal selection matrices for this operation on B and C.

B. Compact Modeling Flow based on Combined Terminal and Model Order Reduction

After the terminal reduction, we will apply the traditional model order reduction technique on the terminal-reduced circuit as shown in Fig. 3. The proposed terminal reduction can work with any existing model order reduction techniques like projection based methods [9], TBR based methods [10] or hierarchical reduction methods [15] etc.

To use the terminal and order reduced models, we may need to build simple interface circuits to connect the original terminals to the representative terminals. For output terminals, we may just physically connect the representative terminals to all the fan-out nodes that the reduced terminals connect before (i.e. no interface circuitry is required). For input terminals, to avoid the adverse coupling among different sources with low impedance, we may use controlled sources (for instance, current controlled current sources (CCCS)) to connect several

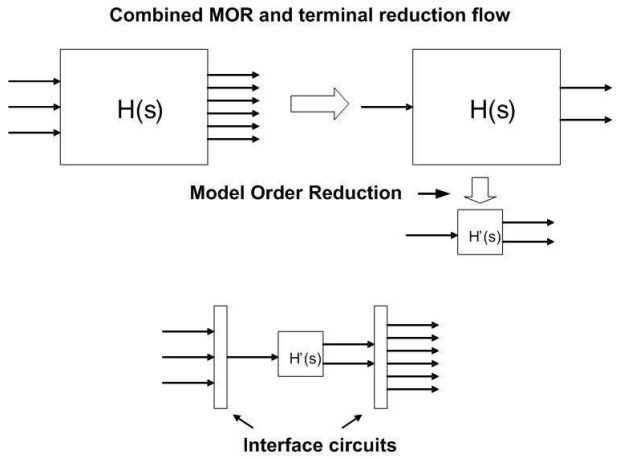

Fig. 3. The reduction flow of combined terminal and model order reductions.

input sources together. For input sources with high impedance, we can just physically connect them together (again, no interface circuitry is required). Fig. 4 illustrates the simple interface circuit for the input and output terminals. The admittances for each CCCS is the corresponding input admittance of the represented input terminal, which can be passively realized using Foster's realization method [13].

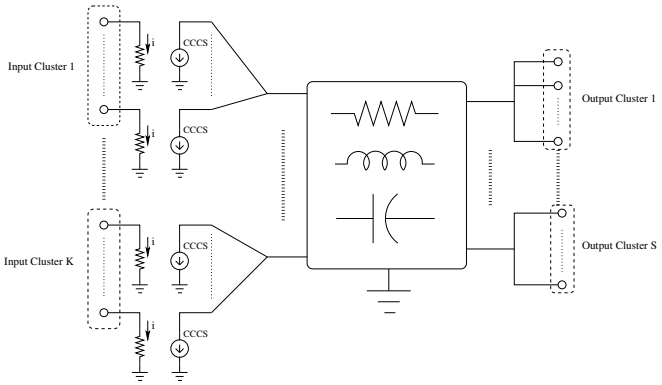

Fig. 4. The illustration of the simple interface circuit.

C. Practical Implementation and Consideration

We have mentioned that we determine the order of moments r based on the fact that the number of moments in the moment series from all the inputs should be equal or large than the number of terminals to be merged as shown in (22) and (23). In this way we can get the complete information of the moment matrix after singular value decomposition. However, if we have many outputs and a few inputs, we end up with large r. In other words, we have to use very high order moments. But high order moments actually are not very informative as they contain mainly the dominant pole information numerically [12]. This is also the case for interconnect circuits with many inputs and a few outputs. But on the other hand, a large number of outputs or inputs does not imply that the circuit has more independent outputs and inputs, or clusters. Practically the final cluster number may be still very small compared to the number of outputs and inputs. In this case, we do not need many high order moment information to distinguish those terminals.

We pre-define an effective cluster number q_e, which is smaller than the number of terminals q to be merged but typically is larger than the resulting number of clusters. For example we may define $q_e = \lceil q/i \rceil, i = 2, 3, \ldots$ depending on circuits, where the function $\lceil x \rceil$ means rounding the x to the nearest larger integer. Then the order of moments r would be equal or large than q_e/p. If the cluster's number is equal to q_e after SVD method at some threshold, that means the pre-defined number of clusters is too small to find the optimal cluster number at this threshold. We then either increase the threshold value (at cost of more approximation errors) or increase the pre-defined clusters' number q_e to re-cluster the terminals. If we already have had some pre-knowledge about the circuit terminals, it will be very helpful to choose the effective cluster number q_e.

To deal with circuits with very large number of terminals, we can apply a hierarchical terminal reduction scheme based on the proposed method. Specifically, we first perform the bottom-up clustering of terminals into a number of groups based on a few moments (or just first order moments, which are Elmore delay). After this we perform SVD on each group and find several representative terminals from each group by K-means clustering method. After this we perform SVD on all the representative terminals to find the best global cluster number. Then we perform K-means algorithm on the top of those selected representative terminals to find the global representative terminals. We may have several hierarchical levels when the number of terminals is very large.

Another issue is that there may exist the DC paths from the inputs to the outputs. In this case, (16) will be written as

$$H(s) = C(sI - A)^{-1}B + D$$

and the zeroth moment will become $\mathbf{m}_0 = -CA^{-1}B + D$. If the zeroth moments for those DC-coupled terminals are quite different, they will unlikely be clustered together. So our method can still be applied.

VII. Passivity Analysis

Passivity is an important issue for compact modeling. First we show that the proposed TermMerg terminal reduction algorithm can always guarantee the passivity of reduced models when the model order reduction is passive regardless of the input and output position matrices B and L. We have the following result:

Proposition 1: The TermMerg method can always lead to passive models for any RLCK circuits.

The proof is obvious. For a any given RLCK circuits with sets of input and output terminals, after the TermMerg reduction, we have a new set of input and output terminals. To make the terminal reduced system passive, one can use two reduction approaches. One is by means of the projection based method. In this case, one first makes all the remaining terminals as bidirectional ones (they are both input and outputs at the same time). As a result, we have the same terminal reduced input and output position matrices. In this case, projection based method can generate the passive model for this circuit.

The second method is by means of passive Truncation Balanced Realization (TBR) method, which can make passive reduction of RLCK circuits with different B and L^T directly but at higher computation costs [11].

As we can see, as long as the model order reduction is a passive process, the combined terminal and model order reduction process is passive for the new method.

However this is not the case for SVDMOR as SVDMOR is highly coupled with the model order reduction process. We analyze two cases. First we have $B \neq L^T$. In this case, B_r and L_r^T are not identical and V_k and V_k are not identical either. As a result, projection based MOR method can't produce passive models. This is also true for passive TBR method as V_k and V_k are not identical.

One may think that one can make all the terminals as bidirectional ones by making B and L^T equal. But we show below such a simple strategy for enforcing passivity may significantly reduce the terminal reduction qualities.

Table I lists the singular value results after the SVD on the 0th and 1th moment matrices for circuit *net1026* with both input and output terminals treated as bidirectional ones.

TABLE I

THE SINGULAR VALUES OF THE DC ADMITTANCE MOMENT, 1TH ORDER ADMITTANCE MOMENT MATRICES OF THE CIRCUIT *net1026* WHEN ALL THE TERMINALS ARE TREATED AS BIDIRECTIONAL ONES.

♯	$\mathbf{m_0}$	$\mathbf{m_1}$
1	0.58970	0.42268
2	0.55093	0.33515
3	0.50215	0.30440
4	0.41875	0.28897
5	0.31229	0.25942
6	0.064575	0.24131
⋮	⋮	⋮
258	0.0023181	0.0099492
259	0.0013246	0.0099117
260	0.00059548	0.0098789
261	0.0001499	0.0098567
262	2.9535×10^{-17}	0.0096795

From the Table I, we can see that the singular values decay very slowly and the terminals can't be reduced very much. Actually the numerical rank of the moment matrix $\mathbf{m_i}$ (given by Matlab rank command) is 261 for $\mathbf{m_0}$ and 262 for $\mathbf{m_1}$, which is the full rank of the moment matrix. This is true for other test cases. As a result, we can observe that SVDMOR is not efficient for reducing bidirectional terminals. One obvious reason is that many output terminals now become input terminals. So the responses excited by those terminals have to be considered for all the other terminals, which make the reduction more difficult or almost impossible.

In the second case, we consider $B = L^T$. To ensure passivity, we requires that B_r are L_r^T are the same and V_k and V_k are identical as well due to the requirements of the congruence transformation. As a result, the moments $\mathbf{m_i}$ must be symmetric. It can be proved that when the inputs to the original models consist of only current sources or voltage sources for RLCK circuits, $\mathbf{m_i}$ is symmetric. If both current and voltage sources are present, $\mathbf{m_i}$ will not be symmetric and terminal reduction by SVDMOR will not ensure the passivity. But as we already have showed that $B = L^T$ will lead to terminal of reduction of bidirectional terminals, which can't be reduced effectively by SVDMOR.

In short, we observe that SVDMOR in general dose not lead to passivity terminal reduction. When it does passive terminal reduction, it does not work well (or not at all). In contrast, TermMerg can always ensure the passive terminal reduction by using a passive model order reduction method.

VIII. EXPERIMENTAL RESULTS

The proposed method has been implemented in MATLAB. We tested our algorithm on a number of real industry interconnect circuits from our industry partner, Cadence Design System Inc.

The first interconnect circuit *net1026* is an one-bit line circuit from a SRAM circuit in 160nm technology. This network contains 525 resistors, 772 capacitors, 6 drivers and 256 receivers. We perform K-means based *TermMerg* algorithm to reduce both receiver (output) terminals and driver (input) terminals. Since there are many outputs(256 receivers) comparing with a few inputs(6 drivers), in this case, we set the order of moments in the input moment matrix is 1 and the order of the moments in the output moment matrix should be 256/6 = 42. But in this case we set the effective cluster number $q_e = 30$. The output moment order is $r = q_e/6 = 5$.

By using the singular value decomposition, the optimal number of clusters is found to be 5 if we define threshold $\varepsilon = 10^{-3}$. The dominant singular values for the M_I, M_O are listed in the Table II.

A. Comparison with the SVDMOR method

One problem with the SVDMOR method is that only a specific moment is used to determine the correlations of both input and output terminals. The specific moment, like DC moment, can give your the good estimation of terminal correlations on the low frequencies, but they may not be accurate at the high frequencies. This is specially true for circuits having small number if inputs and large number of outputs or vice versa.

Fig. 5 shows the SVDMOR reduction results for the interconnect circuit, *net1026*, in frequency domain. We perform first the terminal reduction and then Krylov subspace based MOR. We keep up to 2nd order of block moments in the MOR for both the SVDMOR and TermMerg methods. SVDMOR based on $\mathbf{m}_0$ reduces the terminals to only one input and one output terminals based on the singular values as shown in Table II. For TermMerg, we have one input and five outputs after terminal reduction.

From Fig. 5 we can see that result from SVDMOR does not match well with the original circuit at high frequencies while the proposed method have a better matching at the same high frequencies.

Accuracy loss at high frequency after terminal reduction reflects the fact that the only specific order of moments (DC moments) are used during the SVD-based terminal reduction and DC moments will lead to less accurate terminal relationship at the high frequency ranges. This problem can be easily seen when the numbers of inputs and output terminals are

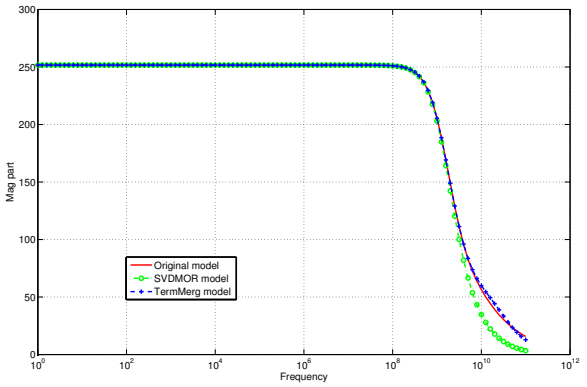

Fig. 5. Frequency impedance responses from the SVDMOR method for *net1026* circuit.

TABLE II

THE SINGULAR VALUES OF DC ADMITTANCE MOMENT, INPUT MOMENT MATRIX AND OUTPUT MOMENT MATRIX OF THE CIRCUIT *net1026*.

♯	$\mathbf{m_0}$	M_I	M_O
1	5789.5	5789.5	46355
2	1.4364×10^{-12}	1.4364×10^{-12}	666.95
3	3.7177×10^{-13}	3.7177×10^{-13}	22.558
4	–	–	0.32084
5	–	–	0.00266
6	–	–	1.4549×10^{-5}
7	–	–	5.6469×10^{-8}

quite different as in case of *net1026*. For this circuit, we can compare 256 (DC) response at 256 output terminals for each input terminal, so we have a better picture for determining their correlations even using the DC moments. While for each output port, we can only compare the responses from 6 inputs, the terminal relationship will not be accurate with just DC moment. Once we use high order moments, we obtain better correlations of the output terminals and thus reduced model is more accurate at the high frequencies as shown in Table II and Fig. 5.

B. Clustering Results

The final clustering results from TERMMERG are shown in Table III. The first column is the number of cluster series. The second column is the representative terminal of each cluster. And all the terminals in each cluster are placed in the third column.

TABLE III

THE OUTPUT CLUSTERING RESULTS FOR THE ONE-BIT LINES CIRCUIT *net1026*.

Cluster #	Rep. Terminal	Clustered Terminals
1	Rcv206	Rcv151, Rcv152, . . . , Rcv256
2	Rcv58	Rcv39, Rcv40, . . . , Rcv77
3	Rcv19	Rcv1, Rcv2, . . . , Rcv38
4	Rcv98	Rcv78, Rcv79, . . . , Rcv119
5	Rcv144	Rcv120, Rcv121, . . . , Rcv150

Fig. 6 plots the distribution of terminals (x-axis) with respect to the cluster index number(y-axis).

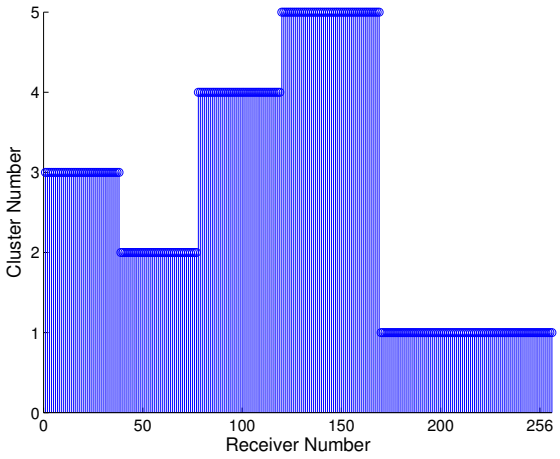

Fig. 6. Output terminal distribution for each cluster for *net1026* circuit.

Then we go back to time domain to validate the effectiveness of our method. We add a voltage source to a driver input to view the step responses at other receivers. Fig. 7 shows the responses of five representative terminals. If we compare the 50% delay time, the delay time difference among them is approximately 10-20ps, which is quite different. The enlarged local waveforms in Fig. 7 are shown in Fig. 8.

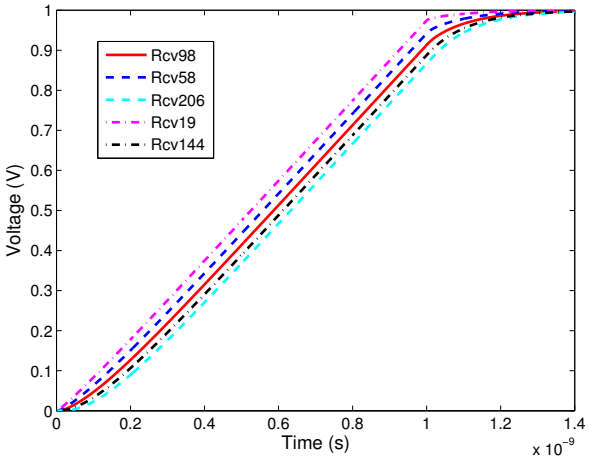

Fig. 7. Step responses of representative output terminals.

If we plot more responses for all the suppressed terminals in one cluster, for instance *receiver97* and *receiver99*, whose representative terminal is the *receiver98*, we can not tell the difference in responses between these reduced terminals and their representative terminal, *receiver98*, for the delay time as shown in Fig. 9. Detailed analysis shows that the delay time differences among these terminals are only about 1-2ps, which is clearly shown in Fig. 10. In other words, if we allow 1-2ps delay variations, those terminals can be viewed as the same terminal.

At this point, we can say that it is reasonable using the response at *receiver98* to represent the responses at the

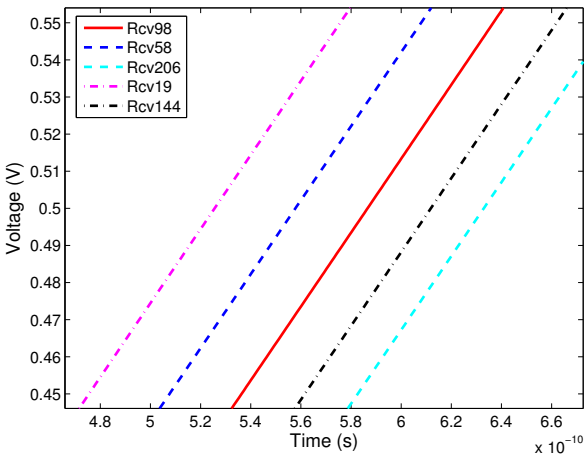

Fig. 8. Comparison of 50% delay time among the representative output terminals.

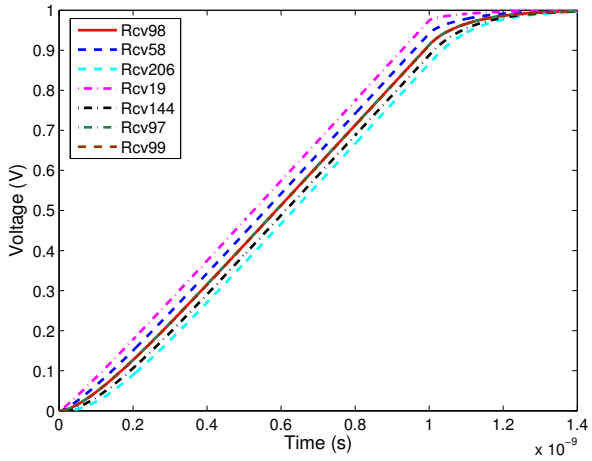

Fig. 9. Step responses of representative output terminals and two suppressed outputs.

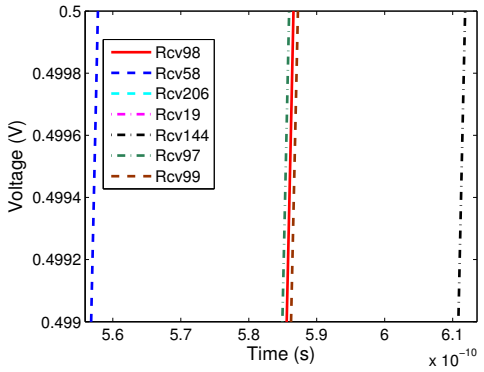

Fig. 10. Comparison of 50% delay time among the representative output terminals and two suppressed outputs.

suppressed terminals in its cluster such as *receiver97* and *receiver99*. Considering the process variations and other environmental variations, it is possible that we can combine them into one terminal. Also we can improve the accuracy of this method by relaxing the threshold ε to generate more number of clusters.

The second example, *net27*, is a clock tree circuit also in 160nm technology. It contains 167 resistors, 654 capacitors, 14 drivers and 118 receivers. For the output reduction, we set the effective cluster number $q_e = 118/2 = 59$. Then we only need $r = q_e/14 \approx 4$ orders of moments to format the output matrix M_O. After the SVD step, the output moment matrix M_O has the following singular values: $\Sigma = diag(52.58, 16.85, 3.41, 0.48, 0.024, 5.70 \times 10^{-6}, \dots)$. Since there is a big magnitude drop between two singular values 0.024 and 5.70×10^{-6}, it is obvious to select cluster number as $k = 5$ at the given $\varepsilon = 1 \times 10^{-3}$.

We also present its distribution of terminals for different clusters in Fig. 11 when we select cluster number $k = 5$. The representative terminals are *receiver98*, *receiver18*, *receiver110*, *receive36*, *receiver84* corresponding to the cluster from 1 to 5.

Notice that the receiver numbers were not assigned based on closeness between terminals for *net27*. But for circuit *net1026*, it seems that receivers are numbered based on the closeness between them.

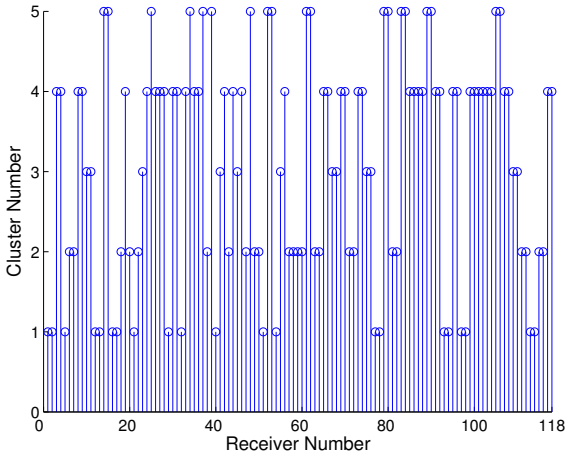

Fig. 11. Output terminal distribution for each cluster for *net27* circuit.

As for the input terminal reduction, the input moment matrix M_I has the following singular values after the SVD: $\Sigma = diag(55.32, 0.091, 1.54 \times 10^{-4}, 8.89 \times 10^{-6}, 8.05 \times 10^{-6}, 6.55 \times 10^{-6}, 6.33 \times 10^{-6}, 6.03 \times 10^{-6}, 5.80 \times 10^{-6}, 6.56 \times 10^{-15}, \dots)$. If we set the threshold $\varepsilon = 1 \times 10^{-3}$, the number of clusters is $k = 10$. However we notice that there is a big drop between 0.091 and 1.54×10^{-4}. If we relax the threshold to $\varepsilon = 0.09$, the cluster number will be only 2. The reduction results will be more efficient but less accuracy. Table IV shows the terminal assignment for each cluster when we cluster terminals at $k = 10$ and $k = 2$. The clustering results are also shown in Fig. 12 and Fig. 13.

TABLE IV

THE INPUT CLUSTERING RESULTS FOR THE CLOCK NETWORK CIRCUIT *net27* AT DIFFERENT THRESHOLDS.

Threshold	Cluster #	Rep. Terminal	Clustered Terminals
$\varepsilon = 0.001$	1	Drv2	Drv2, Drv4
	2	Drv11	Drv11
	3	Drv13	Drv13
	4	Drv10	Drv10
	5	Drv8	Drv8, Drv9
	6	Drv6	Drv6, Drv7
	7	Drv14	Drv14
	8	Drv1	Drv1, Drv3
	9	Drv5	Drv5
	10	Drv12	Drv12
$\varepsilon = 0.09$	1	Drv2	Drv2, Drv4, Drv6, Drv7, Drv8, Drv9
	2	Drv12	Drv1, Drv3, Drv5, Drv10, Drv11, Drv12, Drv13, Dvr14

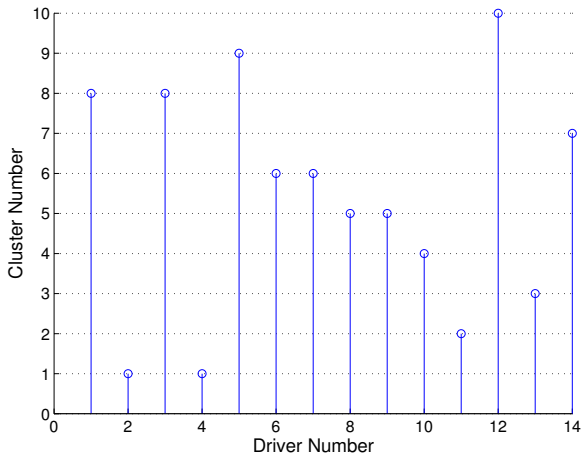

Fig. 12. Input terminal distribution for each cluster for *net27* circuit when $k = 10$.

Fig. 13. Input terminal distribution for each cluster for circuit *net27* when $k = 2$.

IX. CONCLUSION

In this paper, we have proposed a novel method, named **TermMerg**, to efficiently reduce the terminal number of general linear interconnect circuits. We show theoretically that the proposed TermMerg algorithm can always generate passive models for RLCK circuits, which is contrast to SVD-MOR, which does not generate passive models in general. Passivity enforcement in SVDMOR will significantly hamper the terminal reduction qualities. Experimental results on a number of real industry interconnect circuits demonstrated the effectiveness of the proposed method and show also that the proposed method is more accurate than SVDMOR when one the used moment matrix does not give a good terminal correlations.

X. ACKNOWLEDGE

The authors would like to thank Prof. Yinbo Hua at the Electrical Engineering Department in University of California at Riverside for many insightful discussions on singular value decompositions and reviewers for their constructive comments and suggestions which improve the presentation of this paper.

REFERENCES

[1] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. John Wiley and Sons Inc., 2001.

[2] P. Feldmann, "Model order redution techniques for linear systems with large numbers of terminals," in *Proc. European Design and Test Conf. (DATE)*, 2004, pp. 944–947.

[3] P. Feldmann and R. W. Freund, "Efficient linear circuit analysis by pade approximation via the lanczos process," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, no. 5, pp. 639–649, May 1995.

[4] P. Feldmann and F. Liu, "Sparse and efficient reduced order modeling of linear subcircuits with large number of terminals," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, 2004, pp. 88–92.

[5] R. W. Freund, "SPRIM: structure-preserving reduced-order interconnect macromodeling," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, 2004, pp. 80–87.

[6] G. H. Golub and C. V. Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins University Press, 1996.

[7] S. Lee and M. H. Hayes, "Properties of the singular value decomposition for efficient data clustering," *IEEE Signal Process Letters*, vol. 11, no. 11, pp. 862–866, Nov. 2004.

[8] P. Liu, S. X.-D. Tan, H. Li, Z. Qi, J. Kong, B. McGaughy, and L. He, "An efficient method for terminal reduction of interconnect circuits considering delay variations," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, Nov 2005, p. to appear.

[9] A. Odabasioglu, M. Celik, and L. Pileggi, "PRIMA: Passive reduced-order interconnect macromodeling algorithm," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, pp. 645–654, 1998.

[10] J. R. Phillips, L. Daniel, and L. M. Silveira, "Guaranteed passive balancing transformations for model order reduction," in *Proc. Design Automation Conf. (DAC)*, 2002, pp. 52–57.

[11] ——, "Guaranteed passive balanced transformation for model order reduction," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 8, pp. 1027–1041, 2003.

[12] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, pp. 352–366, April 1990.

[13] Z. Qi, S. X.-D. Tan, H. Yu, L. He, and P. Liu, "Wideband modeling of RF/analog circuits via hierarchical multi-point model order reduction," in *Proc. Asia South Pacific Design Automation Conf. (ASPDAC)*, Jan. 2005, pp. 224–229.

[14] M. Silveira, M. Kamon, I. Elfadel, and J. White, "A coordinate-transformed Arnoldi algorithm for generating guaranteed stable reduced-order models of RLC circuits," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, 1996, pp. 288–294.

[15] S. X.-D. Tan, "A general s-domain hierarchical network reduction algorithm," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, 2003, pp. 650–657.