

A Dynamic Capacitance Matching (DCM)-based Current Response Algorithm for Signal Line RC Network

Zhoujie Wu , Cai Luo  and Zhong Guan 

Abstract—This paper proposes a dynamic capacitance matching (DCM)-based RC current response algorithm for calculating the current waveform of a signal line without performing SPICE simulation. Specifically, unlike previous method such as CCS model, driver linear representation, waveform functional fitting or equivalent load capacitance, our algorithm does not rely on fixed reduced model of both standard cell driver and RC load. Instead, our algorithm approaches the current waveform dynamically by computing current responses of the target driver for various load scenarios. Besides, we creatively use symbolic expression to combine the y-parameter of RC network with the pre-characterized driver library in order to perform capacitance matching by considering over/under-shoot effect. Our algorithm is experimentally verified on 40nm CMOS technology and has been partially adopted by latest commercial tool for other nodes. Experimental results show that our algorithm has excellent resolution and promising efficiency compared with traditional methods and SPICE golden result, especially for application in computing delay, power and signal line electromigration.

Index Terms—RC network, symbolic expression, dynamic capacitance, current response, algorithm.

I. INTRODUCTION

IN advanced technology, the parasitic effect of interconnect lines leads to an increasingly large scale RC circuit. It is necessary for electronic design automation (EDA) tools to efficiently and accurately complete complex design tasks. The RC current response of signal lines serves as a crucial parameter for various analyses, including timing, power, and signal line electromigration (EM) reliability analysis. While transistor-level simulators can provide results of gold-standard precision, their use in modern IC design is hindered by memory and time constraints. Consequently, it is common practice to abstract gate-level circuits and simplify the analysis through modeling.

Signal lines, which behave as RC circuits, become progressively more nonlinear as their aspect ratio and length increase, posing analytical challenges. For current estimation in RC circuits, several well-known algorithms based on moment matching such as and AWE [1] and PRIMA [2] have been used for decades. However, they require a known input either in a form of applied current or voltage waveform to predict the current response of the RC circuit. In the case of gate-level

driving, this input also depends on the gate slew, making these algorithms less efficient, so the need to model the gate cell is unavoidable. Methods for timing modeling of gate cells can be broadly categorized as follows.

One approach is the Current Source Model (CSM). Criox and Wong introduced a gate cell current source model called Blade [3], comprising a voltage-controlled current source, internal capacitance, and a one-step time-shift operation. This model effectively simulates the electrical behavior from the input side to the output side of gate cells. Kellor further improved model accuracy by introducing the KTV model, which considers Miller capacitance [4]. Subsequently, nonlinear characteristics of capacitance parameters in CSM models have been considered by Li et al. [5] and Fatemi et al. [6], incorporating input and output parasitic capacitances, Miller capacitance between them, and the output current source as functions of input and output voltages. The CSM model has been further developed in [7]–[12] to address issues such as multi-port, sequential cells, and feedback loops. Currently, widely adopted industry methods such as CCS [13] and ECSM [14] involve the establishment of driver and receiver model for each cell. The lookup table (LUT) characterizes the behavior of the cell at different gate slew and output loads, and the current waveform in the LUT is selected by the effective capacitance, with two different input capacitances C_1 and C_2 used to model the nonlinear receiver input transistor capacitance and the Miller effect. These models are independent of the load, allowing them to handle scenarios like input nonlinearity distortions or even non-monotonic behavior caused by crosstalk.

Another approach is Voltage Response Models (VRM), such as Non-Linear Delay Models (NLDM), which utilizes transistor-level simulation and records the corresponding gate delay and output delay of each standard cell under various input slews and output loads, enabling timing estimation by interpolation/extrapolation of LUT. Since employing the total interconnect capacitance C_{total} is overly pessimistic, considering the influence of interconnect resistance, using iterative [15]–[18] or non-iterative methods [19], [20] can identify effective capacitance to enhance accuracy. The former necessitates iterative calculations of C_{eff} until it converges, typically requiring 5 to 10 iterations for convergence, and in cases of unreasonable initial values, incurring significant CPU time overhead. The latter computes an effective capacitance in a single calculation but necessitates a closed-form expression, and although it performs well in delay analysis, it cannot

Zhoujie Wu, Cai Luo and Zhong Guan are with the School of Microelectronics Science and Technology, Sun Yat-Sen University, Zhuhai, 519082, China. Email: {wuzhj53, luoc23}@mail2.sysu.edu.cn, guanzh23@mail.sysu.edu.cn

Corresponding Author: Zhong Guan.

precisely match the output waveform, with slew errors that could be as high as 15% [21]. Beyond these challenges, as technology scaling, the complexity of RC load makes it challenging to perfectly fit the response curve of the RC network, rendering two-piece output [15] or two effective capacitance [18] insufficient for an ideal fit [22].

Apart from these methods, there are approaches that utilize fitting functions to predict output waveform. Since only the driver's input and the topology of the RC circuit are known, the waveform of the driver's output (input of the RC network) is directly modeled as a parameter-based analytical function. For example, the double exponential function is usually applied to model the current responses [23], while Weibull [24] and gamma [25] function are often used to model the voltage responses, as described in (1)-(3). The related parameters are trained by minimizing the error for specific responses such as the response of the driver loaded with a fixed capacitance. Recently, a macromodeling approach based on the inertial delayed Elmore delay (DED) was proposed for fitting gate output in [26]. This approach utilizes SPICE to extract two macromodel parameters of the gate cell under a single capacitance to rapidly approximate the delay and output waveform within an error of 5%, but it fails to predict the initial over/under-shoot, a limitation that becomes more pronounced when the input slope is substantial.

$$I(t) = K \cdot (e^{-t/T_a} - e^{-t/T_b}) \quad (1)$$

$$V(t) = V_{DD} \cdot (1 - \exp((-\frac{t}{\beta})^\alpha)) \quad (2)$$

$$V(t) = V_{DD} \cdot (1 - \frac{\Gamma(n, \lambda t)}{\Gamma(n)}) \quad (3)$$

Building upon our prior work [27], this paper proposes a novel method for solving RC load response waveform based on dynamic capacitance matching (DCM). It models RC load under the influence of logic gate as dynamic capacitance, employing symbolic expression to seamlessly integrate high-order driving point function of RC circuit with the driver pre-characterization library, calculating the values of dynamic capacitance for N segments while considering their interdependencies. Some approximations are used to ensure algorithmic stability. The algorithm predicts the current waveform by utilizing the current responses of fixed capacitances, either from foundry data or through pre-characterization using SPICE simulation if the data is unavailable. In comparison to traditional methods, even for extensive and intricate RC load, our algorithm can rapidly and accurately predict the nondigital behavior of signal lines. This technique excels in three critical aspects: 1) the degree of fit for voltage/current response curves; 2) computation time; and 3) configurability.

The remainder of this article is organized as follows. In Section II, we provide a relatively detailed explanation of the model order-reduction (MOR) used in our method. In Section III, we explain the theoretical basis of the proposed method and show the complete flow of the algorithm. In Section IV, we compare the simulation results of the classical and the proposed methods. Finally, we conclude in Section V.

II. TYPICAL SIGNAL LINE RC MODEL

Most signal lines can be modeled as pure RC networks. According to the characteristics of linear circuits, researchers are usually interested in their order reduction models. MOR is divided into two categories: One is based on time domain. Typical methods include Chebyshev polynomials and Laguerre polynomials [28]–[31]. A typical feature is that the state variables of the system are expanded by orthogonal polynomials in the time domain, and then the corresponding coefficient matrix of the orthogonal polynomials is used for projection order reduction.

The other is based on frequency domain technology and goes in two directions:

Moment Matching. AWE [1] is the first model applied to electronic circuits. By calculating the system moment explicitly and the coefficient of the transfer function of the reduced order system by means of moment matching, it can quickly analyze the reduced order interconnection system. However, AWE algorithm has the problem of numerical instability, which is mainly due to the power iteration of matrix in its moment calculation process. To solve the numerical stability problem of the explicit moment calculation of AWE algorithm, implicit moment matching methods based on Krylov subspace projection have been proposed successively. Typical algorithms include PVL based on Lanczos process [32], [33] and Arnoldi process [34]. Then, aiming at the passivity problem of reduced order system, PRIMA based on Arnoldi process is proposed. It has many advantages, but it does not maintain the reciprocity of interconnected circuits well. In order to solve this problem, MOR based on structure preservation have been proposed successively [35], [36]. These methods are based on block space projection to maintain the block structure of the reduced order system, typically PRIM algorithm [37], which is mainly based on the characteristics of MNA state matrix for block structure segmentation projection. At the same time, on the basis of first-order system order reduction, the second-order projection or moment matching has also been successfully developed, such as ENOR [38] and SAPOR [39]. In short, MOR based on moment matching mainly includes two parts. The first part is the state transformation. In the transformed state space, the state variables can be sorted according to the important characteristics of the circuit measurement. The second part is to cut off the least important state variables, so as to achieve the reduction of space dimension.

Truncated Balanced Realization (TBR). TBR was first introduced by Moore [40] to describe some states in a system that are difficult to reach and observe. The first-order Truncated Balanced methods applied in the control field include Lyapunov Balance method [40] and random Balance method [41]. In order to solve the problem of Lyapunov equation, a large number of Gramian approximation methods have been proposed [42]–[44], which mainly use the principal subspace method of approximating Gramian to speed up the solution of Gramian equations, such as PMTBR [42]. The essence of the truncated balanced method is a kind of energy balance realization, so the balanced order reduction method should first make some realization of the original system, such as singular

value realization and balanced realization, and then preserve some main characteristics to truncate.

The most mature mainstream approach is the MOR based on moment matching. As our algorithm will utilize the results of moment matching algorithms, we describe the method presented in [2] below.

Using the Modified Nodal Analysis (MNA) circuit state equation representation, a linear circuit with time as variable is expressed by the first-order differential equation as (4).

$$\begin{cases} C\dot{x}_n = -Gx_n + Bu_N \\ i_N = L^T x_n \end{cases} \quad (4)$$

where vector $x_n \in \mathbb{R}^{n \times 1}$ represents the state variable composed of the node voltage, the inductance and the new current introduced by the current source. $G, C \in \mathbb{R}^{n \times n}$ are extracted from the parasitic parameters of the interconnection line, G represents the contribution of conductance, and C represents the contribution of capacitance and inductance. The u_N and i_N vectors denote the port voltages and currents. $B, L \in \mathbb{R}^{n \times N}$ are the input and output incidence matrices.

Let $A = -G^{-1}C$ and $R = G^{-1}B$, so the admittance matrix of the system becomes (5). The utilization of sparse matrix solvers (such as Sparse LU) can significantly expedite this process.

$$Y(s) = L^T (I_n - sA)^{-1} R \quad (5)$$

By employing the Block Arnoldi algorithm, (6) is obtained.

$$\begin{cases} colspan(X) = Kr(A, R, q) \\ X^T A X = H_q \\ X^T X = I_q \end{cases} \quad (6)$$

The function of Block Arnoldi algorithm is to convert system matrix A into upper Hessenberg matrix H_q . Due to the specific form of the upper Hessenberg matrix, the matrix inversion process in equation (6) becomes notably expeditious. Let $x_n = X \cdot z_q$, where $z_q \in \mathbb{R}^{q \times 1}$ is the variable of the reduced order system, give the equations (7) and (8).

$$\begin{cases} H_q \dot{z}_q = z_q - X^T R u_N \\ i_N = L^T X z_q \end{cases} \quad (7)$$

$$Y(s) = L^T X (I_q - sH_q)^{-1} X^T R \quad (8)$$

Eigendecomposition is used to calculate the y -parameter of the reduced order system. After the poles and residues are found, it can be expressed as (9), where q is the number of moments to match in the reduced RC model. It is necessary to choose a suitable matching order between accuracy and speed, [45] select the order of the driving point model based on bandwidth estimation.

$$Y(s) = \sum_{i=j}^q \frac{res_j}{1 - \frac{s}{pole_j}} \quad (9)$$

In short, with the application of moment matching techniques, the driving point function of RC load can always be represented in the form of (9). A higher-order driving point

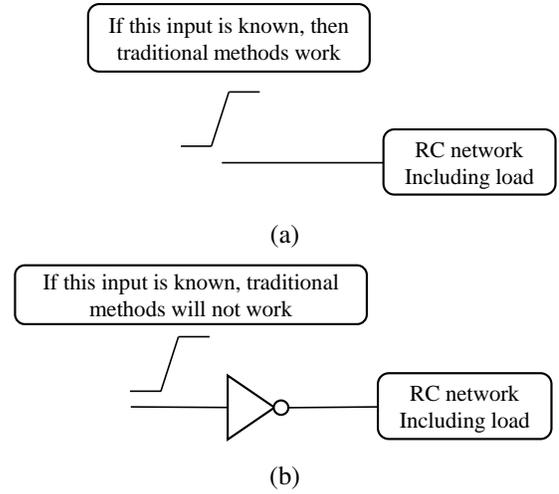


Fig. 1. Limitations of traditional algorithms for RC modeling of signal lines.

function provides results closer to distributed RC than the π -model. When the input to the RC network is known, accurate current/voltage response predictions can be obtained using existing algorithms, as illustrated in Fig. 1 a.

The inputs of a signal line are not always constant and their waveforms strongly depend on both the characteristics of the driver and the load. In response to any change in the driver or the RC load, the current will immediately change, making the algorithms like PRIMA less effective since the current waveform at the driver's output which is the input of the RC network is unknown and needs to be simulated, as depicted in Fig. 1 b.

III. DCM-BASED CURRENT WAVEFORM ESTIMATION ALGORITHM

A. Basic Assumptions and Theoretical Basis for the Proposed Algorithm

Since the signal line current waveform involves both the driver and the load, it is less effective to use traditional approaches to predict the exact current amplitude without performing transistor level simulation. We propose a novel approach for computing the current response of signal line RC networks, enabling precise prediction of voltage and current waveforms.

We notice that all traditional algorithms compute the current response by either simplifying the driver model or the load model. But, regardless of which part is simplified, it is impossible to maintain a consistently accurate current response. Deviations in waveform in one aspect can lead to incorrect conclusions. An ideal approach would require both the driver and load characteristics to be taken into consideration. Recall that moment matching algorithms can efficiently compute the RC current response as long as the input waveform information is given; the challenge becomes how to obtain the input waveform of the RC load, which is the output of a driver. Indeed, without performing transistor level simulation, it is unlikely to capture the exact current waveform. Some additional information is required to bypass the transistor level

simulation. We take advantage of the driver characterization data that is commonly supplied by the foundry.

We develop our method based on the assumption that current (or voltage) responses to the known inputs applied to the driver loaded with different purely capacitive loads are known. Foundries commonly provide them. In case they are not available, SPICE-based pre-characterization using fixed capacitors is required. As long as this information is given, the driver's current response can be determined. This can be illustrated using the current capacitance relation. For any RC load driven by a driver, the current can always be computed using the following (10).

$$I = C \cdot \frac{dV}{dt} \quad (10)$$

$$I(t_n) = \sum_{i=1}^n C \cdot \frac{dV_i}{dt_i} \quad (11)$$

For the same driver, if the environmental parameters (V_{dd} , temperature, etc.) do not change, its output current and voltage value remain identical for the same capacitance load at the same time. This means that the actual current response of a complicated RC load can be seen as a superposition of current responses of multiple pure capacitances in different time frames, as expressed by (11).

To demonstrate the driving capability and to perform delay calculation, foundries provide the current response information as a time versus current amplitude table for each driver loaded with specific capacitive loads. Furthermore, interpolation (or/and extension) fitting functions are employed to describe the transient behavior.

The characterization library saves a small number of driving characteristic curves under various Vdd and temperature, allowing for the interpolation of pre-characterized waveforms at runtime according to the set environment. This ensures that the waveforms produced are highly precise and tailored to the specific design requirements.

If the driver characteristic data provided by the foundry are not available, we can obtain the necessary data performing SPICE simulations. But, this driver RC response characterization must be done in limited simulation runtime. Otherwise, performing a direct transistor level SPICE simulation on the target RC network would be much easier and more convenient. In order to limit the total runtime to characterize the RC response, we set a finite range of the capacitance load and specify its resolution. For example, for a specific driver, we approximate its maximum load as the sum of lump capacitances in its own RC network, recorded as C_{max} . The resolution of 5% of C_{max} is accurate enough for RC characterization and limits the total number of simulations to 20. To characterize those capacitance values below the 5% of C_{max} or between the two reference values, an interpolation technique is used.

B. The Algorithm and its Implementation

The approach we propose here replaces the original driver and input by the voltage curve interpolated (or/and extended)

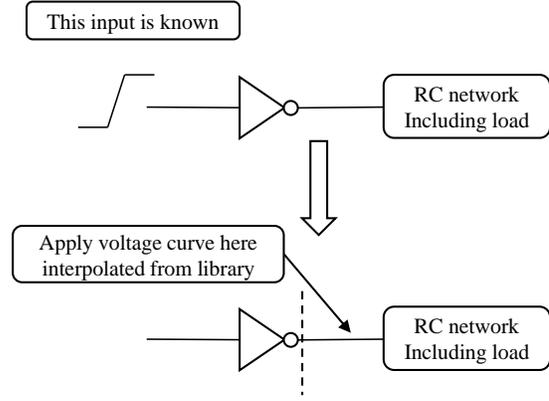


Fig. 2. High level view of the proposed approach. Replace the driver and its input with the voltage curve interpolated from the driver characterization library.

from the driver response table, and then calculates and validates the response using the existing data in the library, as shown in Fig. 2.

In industry, current responses for a specific driver are typically provided as a 3D table showing the current response value versus time and load capacitance.

When we interpolate the discrete current table data as a piecewise function and integrate into voltage, we obtain a voltage response versus time function $V(t)$ for different capacitive loads for the specific driver, as illustrated in Fig. 3, it shows a voltage versus time plane for the driver with modified capacitive load. Different loads result in different voltage response curves in this plane. The output voltage curve of an RC circuit, which is a function of time, is also plotted on this plane. However, the actual curve will be completely different than any curve that maintains a fixed capacitance. Since the effective capacitance of an RC circuit changes over time, the actual curve crosses multiple curves of fixed capacitance until the effective capacitance saturates as the voltage reaches V_{dd} . Whenever the output voltage curve crosses one of the capacitance curves, the time point and output voltage of these two curves are identical. At that time, the capacitance value is the effective capacitance of this RC circuit and the voltage value is the same as the actual voltage of this RC circuit. Therefore, the output voltage of any RC network can be treated as a curve that is obtained by connecting points on different curves of single capacitance responses. In other words, we are constantly changing the value of the effective capacitance, its value is related to time, the driver and the RC network, showing a dynamic in the process.

In order to predict the output voltage without conducting a simulation, we divide the output curve into several regions and determine the average capacitance load associated with each region. Since the driver characterization data is discrete, the average capacitance may not be an exact value in the table. To generate the required $I(t)$ or $V(t)$ for the given capacitive load, interpolation or extrapolation is performed. Our experiments have shown that linear interpolation is accurate enough as long as the resolution of the original table is relatively high (Eg: 5% of C_{max}).

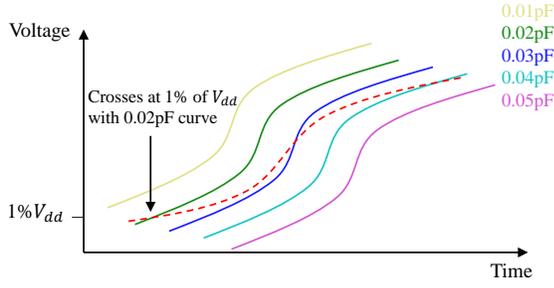


Fig. 3. Voltage versus time plane. The black curves are interpolated from data library as the output voltage responses for different capacitance values of a specific driver (Eg: 0.01 pF to 0.05 pF). The red dashed line is the actual voltage response of an RC circuit crossing the black lines at different points.

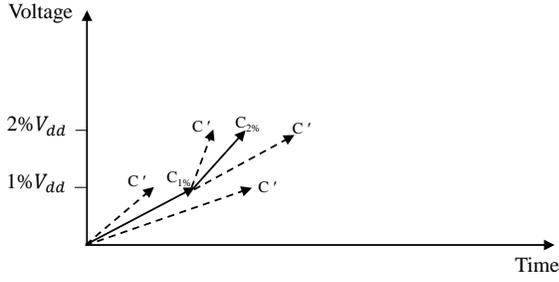


Fig. 4. Searching the effective capacitance for each voltage step based on the driver characterization table.

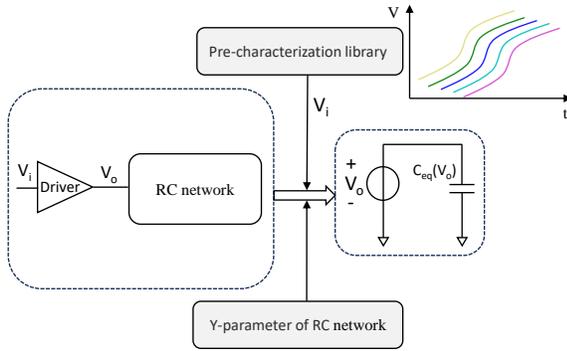


Fig. 5. A (DCM)-based model.

For example, set the number of algorithm segments $N=100$, that is, select 100 equally spaced points. Then the 1% of V_{dd} point of the actual RC response will be on one of the curves (0.02 pF) in the plane with the same 1% V_{dd} , as shown in Fig. 3. Let us assume the 1% of V_{dd} point belongs to the curve with capacitance value $C_{eff,1\%}$ (the effective capacitance when the voltage reaches 1% of V_{dd} , corresponding to 0.02 pF in Fig. 3), then the actual current of the RC circuit will be the same as the current of $C_{eff,1\%}$ when their voltages are both at 1% of V_{dd} . Thus as long as we can find the $C_{1\%}$ curve, we can find the current of the RC circuit when its voltage reaches 1% of V_{dd} .

Starting from 0 to 1% of V_{dd} , we choose one of capacitance curves (C') and apply the voltage source to the RC network. We use an existing algorithm like PRIMA to speed up the

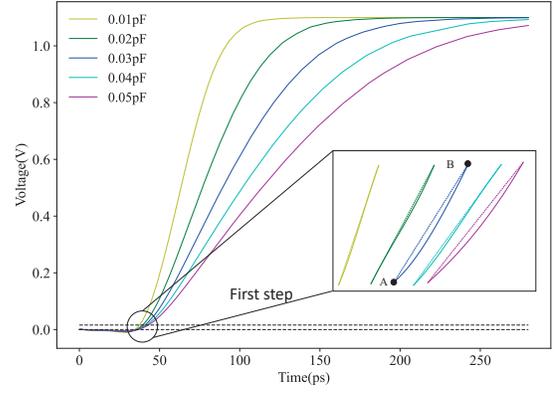


Fig. 6. Driver voltage characterization table. The voltage response curves of different load capacitors are approximated by a straight line if the resolution is sufficient.

calculation of current. Compare the calculated current at 1% of V_{dd} point and the current of C from the table. If these two values match (difference less than tolerance error), we claim that C is the $C_{1\%}$ that we seek. If these two current values do not match, we choose another capacitance curve and repeat this process until we find the correct one, as illustrated in Fig. 4. The DCM-based model is depicted in Fig. 5. It selects a pre-characterized library based on the driver's gate slew. The simplified model consists of a parallel combination of a voltage source and a capacitance. This capacitance, denoted as C_{eq} , is a function of the output voltage, varying among N values, representing the effective capacitance at different points. The more points we select, the more accurate the results become. We will discuss the matching strategy later in the paper.

The key issue is how to quickly and accurately obtain the response current value of the RC network during the matching process, so as to compare it with the library current. We introduce the moment matching algorithm, which simplifies the complex RC network system and obtains the impedance function $Z(s)$ or admittance function $Y(s)$, which is consisting of residue-pole pair. There are two traditional methods to obtain the time-domain current response based on the y -parameter. One is the inverse Laplace transform, as shown in (12), which is time-consuming and has numerical errors. Another way is the convolution integral, as shown in (13), which requires $O(T^2)$ complexity, T is the number of time points during simulation.

$$i(t) = \mathcal{L}^{-1}[U(s) \cdot Y(s)] \quad (12)$$

$$i(t) = \int_0^t y(t - \tau)v(\tau)d\tau \quad (13)$$

These are numerical operations, and only approximate values can be obtained, especially when the RC network is large in scale or small in value. Therefore, we use symbolic expressions to speed things up and eliminate cumulative errors.

When we set the resolution enough (Eg: 1% V_{dd}), its voltage excitation can be replaced by a straight line to obtain approximate current response, similar to the limit, as shown

in Fig. 6. The linear function determined at two points in the voltage pre-characterization curve can be used as part of the piecewise voltage excitation, as shown at points A and B (for 0-1% V_{dd}). We use the piecewise linear (PWL) as the input source of the RC network, according to the fixed form of the y -parameter obtained by the moment matching algorithm, we get a symbolic expression, that is, we only need to perform algebraic operations to get the corresponding current response.

$$v(t) = \sum_{i=1}^{N-1} (k_i(t - t_i) + v_i)u(t - t_i)u(t_{i+1} - t) \quad (14)$$

The voltage PWL is given by (14), where $k_i = \frac{v_{i+1}-v_i}{t_{i+1}-t_i}$, $u(t)$ is the Heaviside step function, and N is the number of segments set by the algorithm, it determines the number of steps (matching points) we need to generate the driver output response. Determine the expression of N-1 segment by continuous iteration, as shown in Fig. 4. The frequency domain expression can be further obtained as (15).

$$\begin{aligned} V(s) &= \mathcal{L}\{v(t)\} \\ &= \int_0^{+\infty} v(t)e^{-st}dt \\ &= \sum_{i=1}^{N-1} \int_{t_i}^{t_{i+1}} (k_i(t - t_i) + v_i)e^{-st}dt \\ &= \sum_{i=1}^{N-1} \left((k_i \frac{1}{s^2} + v_i \frac{1}{s})e^{-st_i} \right. \\ &\quad \left. - (k_i \frac{1}{s^2} + v_{i+1} \frac{1}{s})e^{-st_{i+1}} \right) \end{aligned} \quad (15)$$

Taking the inverse Laplace transform of (9) and (15) yields (16), shown at the bottom of the next page. For a certain residue-pole pair, adding each segment in VPWL gives the partial values, and then the current can be obtained by adding each partial value. After the symbolic expression (16) is obtained through this analysis, the input excitation coordinates (t_i, v_i) and the res-pole corresponding to the RC network can be substituted to obtain $i(t)$ efficiently and accurately. At the beginning of the algorithm, (14) has only one section, and the number of sections is increased until N-1 by finding the matching dynamic capacitance value in iterations, that is, $v(t)$ contains the previously matched excitation section and is used for the calculation of the next section. After the iteration is completed, (14) and (16) are used as the voltage/current waveform of the RC network.

In the search strategy, Since the slope of the voltage response of different load capacitors is monotonic in any step, and there is a unique value C_{eff} in each step. According to (10), when the current value obtained in (16) is smaller than the $i(t)$ in pre-characterized library, slope should be increased, that is, search to the left (decrease capacitance); Otherwise, we need to decrease slope and search to the right. Binary search combined with this strategy can effectively reduce the matching time.

A rapidly rising or falling input waveform will couple to the signal node, causing a reverse current at the output, and

Algorithm 1 DCM-based RC Current Response Computation.

Input: Drivers pre-characterized library; RC load network and the name of its driver.

Output: Response of each driver(voltage/current); AVG, RMS and Peak current.

- 1: **for** each combination of driver and load network **do**
 - 2: Identify the driver's table;
 - 3: Set the parameter N ;
 - 4: Calculate y -parameter of RC network;
 - 5: **for** each step **do**
 - 6: Select the intermediate capacitance in the table to start binary search, interpolate and get coordinates $(t_i, v_i), (t_i, i_i)$;
 - 7: Append the matching results of each previous step to construct the PWL of $v(t)$
 - 8: Calculate $i(t_i)$ according to 16;
 - 9: Compare this value to the table data i_i ;
 - 10: **if** two current values match **then**
 - 11: Record the voltage/current as the actual value of this step;
 - 12: **else**
 - 13: Replace the matching capacitance of this step according to the search strategy, go back to 7;
 - 14: **end if**
 - 15: **end for**
 - 16: Capture its over/under-shoot and tail from the pre-characterized library according to $C_{eff,1}$ and $C_{eff,N-1}$ respectively;
 - 17: Fit the record points for each step;
 - 18: Calculate its AVG, RMS and Peak current;
 - 19: **end for**
-

the node voltage rises above V_{dd} or below V_{ss} , as shown in Fig. 7, the voltage or current is initially overshooted and/or undershooted due to the internal capacitance of driver (such as the overlay/channel capacitance of the MOSFET). This part of the response curve is not monotonous, and it is difficult to solve it by algorithm. In order to quickly obtain this part of the response, the method we adopt is to directly grab the corresponding over/under-shoot curves from the pre-characterized library according to the $C_{eff,1\%}$. The reason for this is that the C_{eff} of the RC network changes gradually when the voltage of the signal line node changes, and can be approximated by the C_{eff} of adjacent steps. Furthermore, in order to ensure convergence at the step of 99% V_{dd} to V_{dd} , the curve corresponding to $C_{eff,99\%}$ in library is connected to the tail of the prediction result. The whole process is summarized in algorithm 1.

IV. SIMULATION RESULTS

To prove the accuracy of the algorithm, we compared the proposed algorithm with previous work. A large range of load conditions and input slope are simulated and the influence of the parameter N is analyzed. Furthermore, we evaluated the runtime of a complete project, including an analysis of several typical signal line benchmarks. We compared the simulation results with traditional methods and SPICE golden results. We

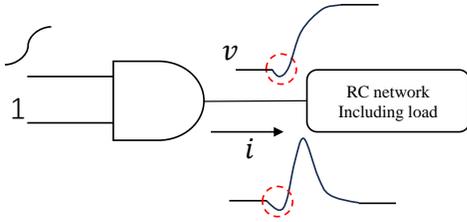


Fig. 7. A rapidly changing input signal induces a reverse current at the output.

TABLE I

THE INTERMEDIATE STEPS OF COMPUTING RC CURRENT RESPONSE FOR THE TEST BENCHMARK.

Time(ps)	Voltage(V)	Current(mA)	$C_{step}(fF)$
0	0	0	0
10.9	-0.026	-0.27	13
16.1	0.016	0.88	13
16.6	0.038	1.07	20
17.3	0.059	1.21	28
18.8	0.102	1.32	36
29.8	0.361	1.22	39
36.9	0.512	1.11	41
44.8	0.650	0.97	43
71.5	0.964	0.43	44
100.7	1.072	0.11	45

assessed the fitting quality of the current waveform using AVG, RMS, and Peak errors. The experiments were conducted using a complete foundry model of a 40nm CMOS process with a 1.1 V power supply.

In Fig. 8, we show the current waveform for a typical clock line benchmark with roughly 2000 resistors and capacitors. The complex RC load of this network has obvious resistance shielding, so the gate output has a large exponential tail. It can be seen that our method achieves very good performance in both current and voltage prediction, well predicting the undershoot waveform in the initial part, leading to a 0.9% error in AVG (1/2 cycle), 0.6% error in RMS and 0.3% error in peak current for this circuit. The total run time is less than 1 second excluding the pre-characterization of the driver.

Table I gives the dynamic capacitance value of each step for the above benchmark during the current estimation. Due to the resistive shielding effect, the effective capacitance is very small at the beginning and gradually saturates until all the capacitances are fully charged.

So as to detect the impact of the number of iteration steps N , we set up different N for analyses. When the input signal changes very quickly (0.01 ns), it will have a negative impact due to insufficient points, as shown in Fig. 9. When N changes from 100 to 70 (not shown) and 50, the current error increases by 0.68% and 4.74% respectively. Fig. 10 shows that for a normal transition time (0.15 ns), $N=50$ is accurate enough. For high-speed circuits with advanced processes, sufficient calculation points ensure accurate analysis results, while analysis points can be reduced for ordinary circuits or non-signoff standards. Calculate according to usage needs.

In Table II, we analyzed a small 40nm CMOS chip containing tens of thousands of signal nets, each consisting of a

TABLE II
THE PROGRAM RUNNING OF THE PROPOSED ALGORITHM UNDER DIFFERENT CALCULATION METHODS (N=100).
EQUIPPED WITH AMD EPYC 64-CORE PROCESSOR @ 2.0GHZ.

Scale	Method	Runtime	Nets per second
24091	Laplace	558 s	43
	Symbolic expression	188 s	128

TABLE III

ERROR OF AVG, RMS AND PEAK CURRENTS AND RUNTIME COMPARISONS (N=100).

	AVG error	RMS error	Peak error	Runtime
CLK(SPICE)	-	-	-	80 s
(C_{eff} model)	20%	10%	2%	<1 s
(Fitting function)	6%	6%	2%	<1 s
(Proposed model)	0.3%	0.1%	0.3%	<1 s
BUS(SPICE)	-	-	-	40 s
(C_{eff} model)	6%	8%	2%	<1 s
(Fitting function)	6%	12%	7%	<1 s
(Proposed model)	1.2%	0.2%	0.5%	<1 s
MUX(SPICE)	-	-	-	16 s
(C_{eff} model)	2%	11%	8%	<1 s
(Fitting function)	17%	1%	4%	<1 s
(Proposed model)	0.8%	0.2%	0.2%	<1 s
SRAM(SPICE)	-	-	-	78 s
(C_{eff} model)	9%	15%	4%	<1 s
(Fitting function)	18%	3%	11%	<1 s
(Proposed model)	0.2%	0.3%	0.2%	<1 s

standard cell and its load RC network. It can be seen that the program using 16 is 3X faster than using the inverse Laplace transform. When N is set to 50, the runtime is reduced by about 10% because part of the program's time is spent solving $Y(s)$.

In order to further validate our algorithm, we simulated several typical representatives of signal line circuits including MUX (load as single driver), Bus-line (simple RC tree with multiple drivers modeled as load) with branches, and SRAM word-line (multiple driver and multiple load drivers). These signal line circuit models contain tens to thousands RC segment and cover the general topology of RC network that signal line is usually modeled with. The results including accuracy and runtime were compared with the NGSPICE circuit simulator results [46], as shown in Table III. The reported runtimes do not include the driver pre-characterization step. If the driver characterization data provided by the foundry are not available, performing such a pre-characterization requires less than 1 minute per driver for 5% of C_{max} resolution. As shown in the table, the proposed algorithm can achieve excellent estimations of current waveforms and the target current values (AVG, RMS and peak) with almost 50-200X faster runtime than NGSPICE on typical signal benchmarks (5-20X faster if we need to pre-characterize driver).

It is worth noting that the proposed algorithm is highly scalable and can be applied to various signal line benchmarks with different levels of complexity, ranging from simple RC trees to complex bus structures and SRAM word-lines. These response waveforms serve as essential parameters for various metrics, including signal line EM reliability, timing analysis,

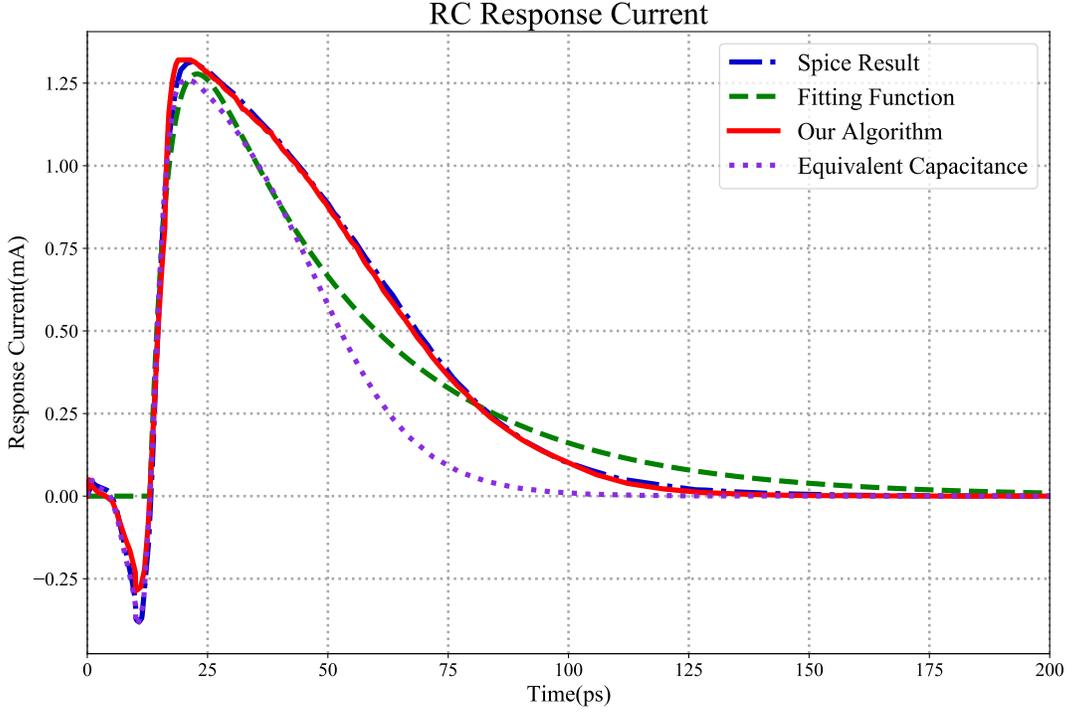


Fig. 8. Current waveforms for the proposed algorithm and previous methods for the test benchmark.

and switching power, among others.

There is a small discrepancy between the results obtained by our algorithm and the golden results of SPICE simulation. The main contributors of the approximation error are the two steps of interpolation/extension (capacitance and voltage/current), Miller effect of the load and finite number of steps we choose. In this paper, we do not show the quantitative error composition. Besides this, there are also several natural limits for the proposed algorithm. First, if the driver is complicated (multi-stage) and the load is simple such that characterization of the driver would be time consuming rather than direct simulation. Second, part of the RC load consists of drivers of next stage that will have a significant Miller effect that

cannot be modeled as fixed capacitance captured in the pre-characterization. Third, the load involves large capacitance so that pre-characterization would be either time consuming or less accurate due to the minimum resolution.

V. CONCLUSION

In this paper, we propose a dynamic capacitance matching (DCM)-based RC current response algorithm for calculating the current waveform of a signal line without the necessity of SPICE simulation is proposed. Unlike previous methods, our algorithm does not depend on function fitting or a single effective capacitance. Instead, it seamlessly integrates the high-order driving-point functions of the RC network with the

$$\begin{aligned}
 i(t) &= \mathcal{L}^{-1}\{V(s)Y(s)\} \\
 &= \sum_{j=1}^q \sum_{i=1}^{N-1} \frac{res_j k_i}{pole_j} ((pole_j(t-t_i) - e^{(t-t_i)pole_j} + 1)u(t-t_i) - (pole_j(t-t_{i+1}) - e^{(t-t_{i+1})pole_j} + 1)u(t-t_{i+1})) \\
 &\quad + \sum_{j=1}^q \sum_{i=1}^{N-1} res_j (v_i(1 - e^{t-t_i})u(t-t_i) - v_{i+1}(1 - e^{t-t_{i+1}})u(t-t_{i+1})) \\
 &= \sum_{j=1}^q \sum_{i=1}^{N-1} \frac{res_j k_i}{pole_j} ((pole_j(t-t_i) - e^{(t-t_i)pole_j} + 1)u(t-t_i) - (pole_j(t-t_{i+1}) - e^{(t-t_{i+1})pole_j} + 1)u(t-t_{i+1})) \\
 &\quad + \sum_{j=1}^q res_j (v_1(1 - e^{t-t_1})u(t-t_1) - v_N(1 - e^{t-t_N})u(t-t_N))
 \end{aligned} \tag{16}$$

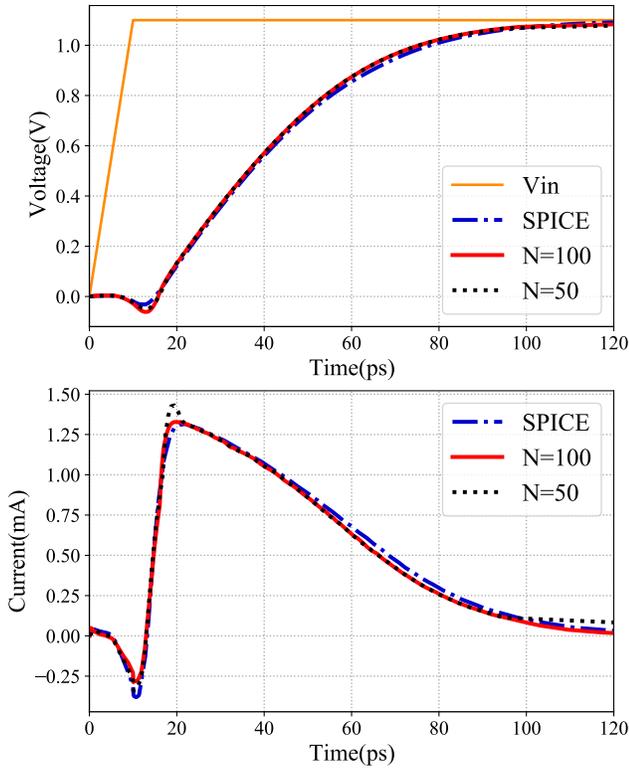


Fig. 9. Apply different N to analyze a steep ramp input.

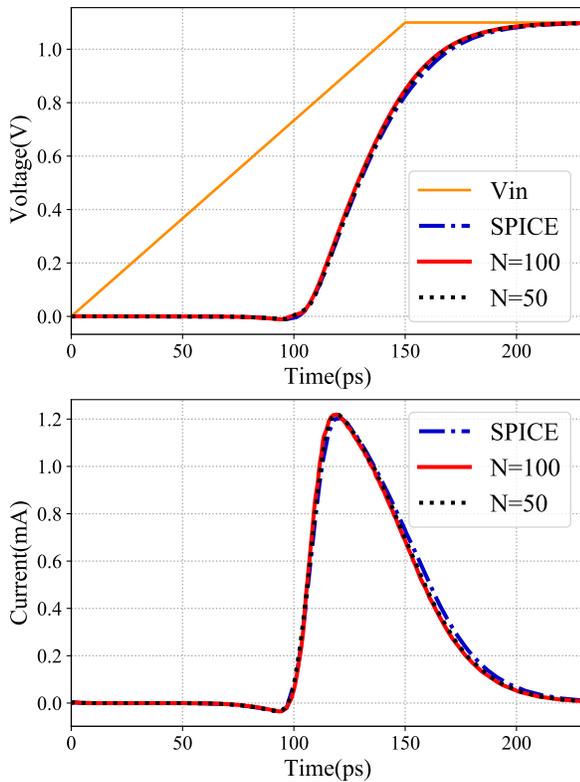


Fig. 10. Apply different N to analyze a normal input.

pre-characterized library of drivers through symbolic expressions, swiftly performing algebraic manipulations to determine

dynamic capacitance values under different output voltages. Dynamic capacitance precisely characterizes the behavior of gate cells under any RC load, exploiting their mutual interdependencies to precisely match the driver’s output current and account for overshoots/undershoots at any given moment. This process is highly configurable, allowing for adjustments tailored to specific application scenarios. Experimental results show that the AVG, RMS and Peak current errors are nearly 1% and the running time is 50-200X faster than the golden results obtained by NGSPICE.

REFERENCES

- [1] L. T. Pillage and R. A. Rohrer, “Asymptotic waveform evaluation for timing analysis,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 9, no. 4, pp. 352-366, Apr. 1990.
- [2] A. Odabasioglu, M. Celik and L. T. Pileggi, “PRIMA: passive reduced-order interconnect macromodeling algorithm,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 17, no. 8, pp. 645-654, Aug. 1998.
- [3] J. F. Croix and D. F. Wong, “Blade and razor: cell and interconnect delay analysis using current-based models,” in *Proc. 40th Annu DAC*, pp. 386-389, June, 2003.
- [4] I. Keller, Ken Tseng and N. Verghese, “A robust cell-level crosstalk delay change analysis,” in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, pp. 147-154, 2004.
- [5] Peng Li and E. Acar, “A waveform independent gate model for accurate timing analysis,” in *Proc. ICCD*, pp. 363-365, Oct. 2005.
- [6] H. Fatemi, S. Nazarian and M. Pedram, “Statistical logic cell delay analysis using a current-based model,” in *Design Automation Conference (DAC)*, pp. 253-256, 2006.
- [7] C. Amin, C. Kashyap, N. Menezes, K. Killpack and E. Chiprout, “A multi-port current source model for multiple-input switching effects in CMOS library cells,” in *Proc. ACM/IEEE Design Automa. Conf. (DAC)*, pp. 247-252, 2006.
- [8] C. Kashyap, C. Amin, N. Menezes and E. Chiprout, “A nonlinear cell macromodel for digital applications,” in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, pp. 678-685, 2007.
- [9] N. Menezes, C. Kashyap and C. Amin, “A “true” electrical cell model for timing, noise, and power grid verification,” in *Proc. ACM/IEEE Design Automa. Conf. (DAC)*, pp. 462-467, 2008.
- [10] B. Amelifard, S. Hatami, H. Fatemi and M. Pedram, “A Current Source Model for CMOS Logic Cells Considering Multiple Input Switching and Stack Effect,” in *Proc. Design Autom. Test Eur. (DATE)*, pp. 568-573, 2008.
- [11] S. Nazarian, H. Fatemi and M. Pedram, “Accurate Timing and Noise Analysis of Combinational and Sequential Logic Cells Using Current Source Modeling,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 1, pp. 92-103, Jan. 2011.
- [12] N. K. Katam and M. Pedram, “Timing Characterization for Static Timing Analysis of Single Flux Quantum Circuits,” *IEEE Trans. Appl. Supercond.*, vol. 29, no. 6, pp. 1-8, Sept. 2019.
- [13] Synopsys—Composite Current Source (CCS). Available online at: <http://www.opensourceilberty.org/ccspaper/>
- [14] Cadence—Effective Current Source Model. Available online at: https://www.cadence.com/en_US/home/alliances/standards-and-languages/ecsm-library-format.html
- [15] J. Qian, S. Pulella and L. Pillage, “Modeling the “Effective capacitance” for the RC interconnect of CMOS gates,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 13, no. 12, pp. 1526-1535, Dec. 1994.
- [16] S. Abbaspour and M. Pedram, “Calculating the effective capacitance for the RC interconnect in VDSM technologies,” in *Proc. ASP-DAC*, pp. 43-48, 2003.
- [17] F. Dartu, N. Menezes and L. T. Pileggi, “Performance computation for precharacterized CMOS gates with RC loads,” *IEEE Trans. Comput.-Aided Design Inter. Circuits Syst.*, vol. 15, no. 5, pp. 544-553, May, 1996.
- [18] J. M. Wang, J. Li, S. Yanamanamanda, L. K. Vakati and K. K. Muchherla, “Modeling the Driver Load in the Presence of Process Variations,” *IEEE Trans. Comput.-Aided Design Inter. Circuits Syst.*, vol. 25, no. 10, pp. 2264-2275, Oct. 2006.
- [19] A. B. Kahng and S. Muddu, “Improved effective capacitance computations for use in logic and layout optimization,” in *Proc. 12th Int. Conf. VLSI Design*, pp. 578-582, 1999.

- [20] M. Shao, M. D. F. Wong, H. Cao, Y. Gao, L. -Po Yuan, L. -D. Huang, and S. Lee, "Explicit gate delay model for timing evaluation," in *Proc. ACM/IEEE International Symposium on Physical Design*, pp. 32-38, Apr. 2003.
- [21] M. Jiang, Q. Li, Z. Huang and Y. Inoue, "A non-iterative effective capacitance model for CMOS gate delay computing," in *ICCAS*, pp. 896-900, July, 2010.
- [22] D. Garyfallou et al., "Gate Delay Estimation With Library Compatible Current Source Models and Effective Capacitance," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 29, no. 5, pp. 962-972, May, 2021.
- [23] P. Jain and A. Jain, "Accurate Current Estimation for Interconnect Reliability Analysis," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 9, pp. 1634-1644, Sept. 2012.
- [24] C. S. Amin, F. Dartu and Y. I. Ismail, "Weibull based analytical waveform model," *IEEE Trans. Comput.-Aided Design Inter. Circuits Syst.*, vol.24, no. 8, pp. 1156-1168, July, 2005.
- [25] Tao Lin, E. Acar and L. Pileggi, "h-gamma: an RC delay metric based on a gamma distribution approximation of the homogeneous response," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, pp. 19-25, Nov. 1998.
- [26] N. Mirzaie and R. Rohrer, "A Macromodeling Approach for Analog Behavior of Digital Integrated Circuits," *IEEE Trans. Comput.-Aided Design Inter. Circuits Syst.*, vol. 39, no. 12, pp. 5025-5031, Dec. 2020.
- [27] Z. Guan and M. Marek-Sadowska, "An efficient and accurate algorithm for computing RC current response with applications to EM reliability evaluation," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, pp. 1-6, Nov. 2016.
- [28] Y. -L. Jiang and H. -B. Chen, "Time Domain Model Order Reduction of General Orthogonal Polynomials for Linear Input-Output Systems," *IEEE Trans. Autom. Control*, vol. 57, no. 2, pp. 330-343, Feb. 2012.
- [29] R. Eid and B. Lohmann, "Moment matching model order reduction in time-domain via Laguerre series," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 3198-3203, 2008.
- [30] J. M. Wang, E. S. Kuh and Qinglian Yu, "The Chebyshev expansion based passive model for distributed interconnect networks," in *Proc. Int. Conf. Computer-Aided Design*, pp. 370-375, Nov. 1999.
- [31] Y. Chen, V. Balakrishnan, C. . -K. Koh and K. Roy, "Model reduction in the time-domain using Laguerre polynomials and Krylov methods," in *Proc. Design, Autom. Test Eur. Conf. Exhibition*, pp. 931-935, Mar. 2002.
- [32] P. Feldmann and R. W. Freund, "Efficient linear circuit analysis by Pade approximation via the Lanczos process," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 14, no. 5, pp. 639-649, May, 1995.
- [33] R. W. Freund and P. Feldmann, "Reduced-order modeling of large linear passive multi-terminal circuits using matrix-Pade approximation," in *Proc. IEEE Des., Autom. Test Eur. Conf.*, pp. 530-537, Feb. 1998.
- [34] B. Salimbahrami and B. Lohmann, "A Two-Sided Arnoldi-Algorithm with Stopping Criterion and an application in Order Reduction of MEMS," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 11, no. 1, pp. 79-93, 2005.
- [35] N. Mi, B. Yan, S. X. . -D. Tan, J. Fan and H. Yu, "General Block Structure-Preserving Reduced Order Modeling of Linear Dynamic Circuits," *ISQED*, pp. 633-638, 2007.
- [36] RC. Li, and Z. Bai, "Structure-Preserving Model Reduction Using a Krylov Subspace Projection Formulation," *Communications in Mathematical Sciences*, vol. 3, no. 2, pp. 179-199, 2004.
- [37] R. W. Freund, "SPRIM: structure-preserving reduced-order interconnect macromodeling," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, pp. 80-87, Nov. 2004.
- [38] B. N. Sheehan, "ENOR: model order reduction of RLC circuits using nodal equations for efficient factorization," in *Proc. IEEE Des., Autom. Conf.*, pp. 17-21, Jun. 1999.
- [39] Yangfeng Su, Jian Wang, Xuan Zeng, Zhaojun Bai, C. Chiang and D. Zhou, "SAPOR: second-order Arnoldi method for passive order reduction of RCS circuits," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, pp. 74-79, Nov. 2004.
- [40] B. Moore, "Principal component analysis in linear systems: Controllability, observability, and model reduction," *IEEE Trans. Autom. Control.*, vol. 26, no. 1, pp. 17-32, Feb, 1981.
- [41] U. Desai and D. Pal, "A transformation approach to stochastic model reduction," *IEEE Trans. Autom. Control*, vol. 29, no. 12, pp. 1097-1100, Dec. 1984.
- [42] J. R. Phillips and L. M. Silveira, "Poor man's TBR: a simple model reduction scheme," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 1, pp. 43-55, Jan. 2005.
- [43] S. Ghosh and N. Senroy, "Balanced Truncation Based Reduced Order Modeling of Wind Farm," *Electrical Power and Energy Systems*, vol. 53, no. 11, pp. 649-655, May, 2013.
- [44] B. David and X. Chen, "Coupled Electrothermal-Mechanical Analysis for MEMS via Model Order Reduction," *Finite Elements in Analysis and Design*, vol. 46, no. 12, pp. 1068-1076, 2010.
- [45] N. Gopal, D. P. Neikirk and L. T. Pillage, "Evaluating RC-interconnect using moment-matching approximations," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, pp. 74-77, Nov. 1991.
- [46] Ngspice Circuit Simulator. Available online at: <http://www.ngspice.org>.