Enhancing Model Order Reduction for Nonlinear Analog Circuits Simulation

Henda Aridhi, Mohamed H. Zaki and Sofiène Tahar
Dept. of Electrical and Computer Engineering, Concordia University, Montréal, Québec, Canada
Email: {h_aridh, mzaki, tahar}@ece.concordia.ca

***Abstract*—Traditionally, model order reduction methods have been used to reduce the computational complexity of mathematical models of dynamic systems while preserving their functional characteristics. This technique can also be used to fasten analog circuit simulations without sacrificing their highly nonlinear behavior. In this paper, we present an iterative approach for reducing the computational complexity of nonlinear analog circuits using piecewise linear approximations, K-means clustering, and Krylov space projection techniques. We show that, using our approach, the robustness of the reduced models is enhanced because of better selection of linearized points obtained through qualitative circuit simulation. We model primary circuit inputs, design initial conditions, and circuit parameters as fuzzy variables with different distributions in qualitative simulations. We then iteratively fine-tune the reduced models until a model is achieved that meets a predefined performance and accuracy conformance criteria. The proposed model order reduction method supports both flat and hierarchical circuit analysis thereby making it suitable for fast and accurate simulation of larger sized highly nonlinear analog, digital and mixed signal circuits. We demonstrate the effectiveness of our method using several key nonlinear circuits used in modern SoC today. They include: a transmission line, a ring oscillator, a voltage controlled oscillator, a phase locked loop, and an analog comparator circuit. Our experiments show that the reduced model simulations are fast and accurate compared to existing techniques.**

***Index Terms*—Model Order Reduction; Analog Circuits; Krylov Space; Qualitative Simulation.**

I. Introduction

Computer simulation is an essential step in the design and verification of analog Integrated Circuits (ICs). Accurate mathematical models of analog circuits tend to be large and thus their computer simulations are computationally expensive in terms of both memory and CPU resources. A large number of computer simulations are generally required to verify their various functional and performance properties. Accurate and less complex mathematical models of analog circuits can help fasten their verification and optimization processes and in meeting the ever pressing time-to-market constraints.

Model Order Reduction (MOR) is a promising technique that reduces the size and complexity of large scale mathematical models while preserving their main characteristics [1]. This technique has been successfully put in practice for the case of linear analog ICs using Krylov space projections and Singular Value Decomposition (SVD) [2]. However, the problem of elaborating a method for the reduction of nonlinear analog circuit models is still an active research area and only a few methods have proven their effectiveness for applications such as nonlinear transmission lines, amplifiers and oscillators [3] [4] [5] [6]. These methods rely on transforming the nonlinear model into a set of local linear or polynomial models and applying projection to obtain reduced local models which can be evaluated on the fly to approximate the original full order model. The idea is very attractive but is based on various heuristics. It is, therefore, not always guaranteed to provide accurate reduced models for analog circuits given that their behavior can be highly nonlinear. Large analog circuits are often synthesized in a hierarchical way resulting in a circuit structure that consists of an interconnection of several instances of linear and nonlinear sub-circuits with possibly different parameters. This fact can be used to help reduce the overall computational cost by independently customizing the MOR parameter for each sub-circuit as described in [7] for linear analog circuits.

Qualitative Simulation (QS) is a method that can be used to characterize the behavior of dynamic systems. It utilizes multivariate optimization techniques in which dynamic systems parameters and initial conditions are considered fuzzy variables with associated possibility distributions [8] instead of concrete values. When applied to analog circuits, it provides the set of trajectories which reach the state variables bounds given all possible variations in their models. Therefore, it offers a better coverage of their reachable state space and characterization of their nonlinear behavior compared with sampling based methods such as the Monte Carlo simulation method [9].

In this paper, we propose a new method to MOR of nonlinear analog circuits. We model analog circuits using fuzzy dynamical models and use QS to characterize and determine important trajectories and envelopes of their state variables. Then, we employ the K-means clustering algorithm to subdivide the circuit state space into discrete regions containing its main responses. In each region, a linearized model is reduced via Krylov space projections [10] and is used to approximate the full order nonlinear circuit model. The number of clusters required by our MOR method is determined through an optimization problem constrained by a minimum behavioral error between the original model and its piecewise linearized model. We also extend the state space by adding the input as a decision variable for the evaluation of the reduced models. Moreover, we establish a set of conformance checking criteria and refine the reduced model to guarantee simulation acceleration and accuracy. We illustrate our proposed methodology on different analog circuits: a transmission line, a ring

oscillator, a voltage controlled oscillator, a phase locked loop, and an analog comparator.

The rest of the paper is organized as follows: first, we give an overview of existing MOR techniques developed for large analog ICs in Section II. Then, in Section III, we briefly explain MOR through projection and qualitative simulation. After that, Section IV details the different steps of our proposed MOR method. Finally, in Section V, we display and discuss our experimental results and in Section VI, we present our conclusions and future work.

II. Related Work

The *state of the art* MOR methods for VLSI circuits are based on Krylov subspace projections and SVD [1]. The Krylov subspace [10] projection methods perform an implicit transfer function moment matching via efficient algorithms like Arnoldi or Lanczos processes [1]. The SVD is a matrix factorization which can be used to compute a low-rank matrix approximation of a given large matrix [11]. These two methods have been used successfully for reducing the size of linear analog circuits differential models, as detailed in [2]. A framework for Hierarchical MOR (HMOR), which rely on partitioning the circuits into blocks, was introduced in [12]. The target application was the reduction of linear interconnect models for signal integrity analysis. In [13], HMOR was used to enhance the performance analysis of large RLKC power delivery systems. Also, in [7] the authors introduced a hierarchical MOR method to reduce large linear blocks of ICs while preserving their passivity.

The above techniques can efficiently address the problem of reduction of linear analog IC models and linearized models, such that RC, RL, and RLC networks. However, the existing MOR techniques for nonlinear circuit models are based on various heuristics and have many limitations related to the dependency on the inputs, the initial conditions variations and the parameters of the nonlinear circuit model under consideration. The Proper Orthogonal Decomposition (POD) [14] is a straightforward application of the SVD reduction method to the case of a nonlinear dynamical system. It uses singular vectors of the system response for a fixed input to project the system dynamics into a smaller state space retaining only its main singular values. In [6], the authors used the POD to derive surrogate models for semiconductors devices that are modeled by Drift-Diffusion Equations. They also outlined the drawback of this method that is the dependency on the original model inputs and parameters.

In [15] and [16], the author presented a simple method for automatically extracting macro-models of weakly nonlinear circuits with time-varying operating points based on Volterra series and variational analysis theory [17]. In [18], the authors outlined that the efficiency of the methods [15] and [16] is limited because of the exponential increase of the size of the Volterra series descriptions. They proposed to enhance them by using a two sided projection method. In [19], nonlinear systems are approximated with quadratic Taylor approximations and reduced via Krylov space projections. This method is accurate only if the nonlinear model is similar to a quadratic approximation. These limitations have been addressed in [20] by rewriting strongly nonlinear models in quadratic-linear form without any approximation. However, this method increases the size of the initial model by introducing new variables and equations and scale poorly with the number of nonlinear terms.

The Trajectory PieceWise Linear (TPWL) [3] MOR method utilizes an aggregation of local linear approximations around expansion points that are selected from the trained original model trajectories. This method was used for the case of weakly nonlinear systems such as nonlinear transmission lines, amplifier chains and Micro-machined devices. The accuracy of the TPWL method heavily depends on the extracted expansion points and the training inputs. An improvement for the linear models aggregation in the TPWL MOR method, that uses state velocities in weights computation, is introduced in [21]. An enhancement of the TPWL MOR method, which consists in an adaptive sampling of the linearization points across the model trajectory based on the error between the nonlinear model and its linearized form, is proposed in [22]. In [5], the authors used a k-means clustering to optimize the set of linearization points and used simple weights to improve speedup of the TPWL method. The main steps of the TPWL MOR method have been followed with a replacement of the local linear models with local PieceWise Polynomial (PWP) models [4] and Tchebychev interpolating polynomial models [23], respectively. This approaches improved the accuracy of the local reduced models but increased their on-the-fly evaluation time. In [24], the authors presented a nonlinear MOR method that constructs parameterized manifolds capturing DC and AC responses for nonlinear systems using symbolic transformations and Krylov projections.

In [25], the authors presented a methodology to approximate nonlinear analog circuit models with a compact set of analytical behavioral models. These models are obtained by extracting nodal matrices from SPICE transient simulations and using recursive vector fitting regression algorithm. This method enhances the MOR method automation but does not overcome the input dependency problem. A similar idea, which consist in using curve fitting to automatically generate analog circuits model from their SPICE transient simulation traces and reducing them using Krylov space projections, is proposed in [26]. In [27], the authors formulated the identification of stable compact models for radio frequency systems as a semi-definite optimization problem. In [28], a method for high-order Volterra transfer functions reduction via Krylov projection is proposed. The novelty of the method consists in using association of multivariate (Laplace) variables in high-order multiple-input multiple-output (MIMO) transfer functions to generate the standard single-s transfer functions.

While the cited MOR methods present a variety of techniques to reduce different types of circuit models, there are still many limitations that need to be addressed. For example the reduced model dependency on the input, the parameters, the number and the selection procedure of the expansion points, and the number and type of weighting functions when using piecewise linear or polynomial approximations have to be addressed. Also, the optimization of the model reduction effort

and the verification system level simulation acceleration using reduced models are still not approached. In this work, we propose to enhance the robustness of MOR methods based on piecewise linear representations by performing qualitative simulation of the circuit behavior and using global optimization to predetermine good guesses of the parameters of the reduction process such as the number and the location of the linearization points. During the on-the-fly integration of the reduced model, we use both the input and the state variables to determine the closest linear models and use simple weight functions to avoid increasing its computational cost. We also present a method to apply the MOR scheme in a hierarchical way to optimize the reduction effort and preserve better the nonlinear behavior of the original model.

III. Preliminaries

A. Model Order Reduction (MOR)

The majority of analog ICs can be described by the large differential model in Equation (1)[1].

$$\begin{aligned} \dot{x} &= f(x,t,u,p) \\ y &= g(x) \end{aligned} \tag{1}$$

where $f : R^n \rightarrow R^n$ is a vector valued function, x and $\dot{x} \in R^n$ are the state vector and its time derivative, t is the time, $u \in R^m$ is a vector of inputs, $p \in R^{n_p}$ is a set of parameters, $y \in R^{n_y}$ is a vector of outputs, and $g : R^n \rightarrow R^{n_y}$ is an output generation function.

MOR consists of transforming algorithmically the large mathematical model $Model(n)$ of Equation (1) to a smaller model $Model(q)$, as given in Equation (2).

$$\begin{aligned} \dot{z} &= \hat{f}(z,t,u,\hat{p}) \\ \hat{y} &= \hat{g}(z) \end{aligned} \tag{2}$$

where $\hat{f} : R^q \rightarrow R^q$ is a vector valued function, z and $\dot{z} \in R^q$ are the state reduced vector and its time derivative, $\hat{p} \in R^{n_{\hat{p}}}$ is a set of parameters, $\hat{y} \in R^{n_y}$ is an approximate of the output vector y, and $\hat{g} : R^q \rightarrow R^{n_y}$ is an output generation function.

The reduced model has a smaller size $q \ll n$, has less parameters $n_{\hat{p}} \leq n_p$, is less computationally expensive than $Model(n)$, and its output accurately reproduces the behavior of $Model(n)$. For the case of analog IC models, the creation of a robust method which outputs, in a finite number of step, an accurate and efficient reduced model is very challenging. It should be customized and elaborated based on the common characteristics of analog IC models which are their infinite state space, their sensitivity to parameters and environment conditions, their nonlinearity, and the type of analysis (DC steady state, AC steady state, transient simulations, etc.). In this paper, we focus on reducing the number of equations required to describe the model while preserving their parameters and input dependency.

[1]All terms defined, in this section, have the same meaning in the rest of this paper, unless stated differently.

B. Projection Based MOR

The MOR of the mathematical model in Equation (1) via projection consists in finding an $n \times q$ unitary projection matrix ($V \cdot V^t = I_n$) and using the projection $\hat{x} = V \cdot z$ as an approximate of the original state vector x, where z is the reduced state vector of variables. There are different algorithms which compute such reduction matrix, e.g., the Arnoldi or Lanczos algorithm [1], the SVD [2], and the POD [6]. For linear dynamical models, only one projection matrix is needed to perform the reduction of the full order model. However, reducing a nonlinear model is extremely challenging. It requires as much reduction matrices as the number of piecewise linear models used to approximate its nonlinear behavior. This is the idea behind the TPWL [3] method and its derivatives [4] [5], as well as the method we propose in this paper.

C. Qualitative Simulation (QS)

QS is an extension of traditional numerical-logical integration methods for dynamical system performance analysis [8]. It uses Fuzzy Differential Equations (FDE), which are differential equations where the deterministic quantities, such as parameters, coefficients, and/or initial conditions, are considered as fuzzy numbers, as given in Equation (3).

$$\begin{aligned} \dot{x} &= f(x,t,u,p) \\ x(0) &= \mu_{x(0)},\ u = \mu_u,\ p = \mu_p \end{aligned} \tag{3}$$

where $\mu_{x(0)}$, μ_u, and μ_p refer to the membership functions used for the possibility distributions of the fuzzy numbers $x(0)$, u, and p, respectively. The membership functions can have uniform, Gaussian, trapezoidal, triangular, or bell function forms. They are transformed into sets of intervals by means of $\alpha - cut$ levels during the QS. Therefore, the uniform membership function is represented with a single $\alpha - cut$ level, the Gaussian membership function allows the evaluation of the model for different $\alpha - cut$ levels and consequently requires larger QS runtime than the uniform membership function [8].

To illustrate the use of QS, we explain its application for the case of a dynamical system subject to uncertain initial state values. The QS finds an envelope of all system trajectories originating in time from the fuzzy number $\mu_{x(0)}$. It formulates this problem as a multi-objective optimization problem, as given in Equation (4).

$$\begin{aligned} \min\ F(y) &= \int_0^{t^*} f(x,t,u,p)\ dt,\ y = x(0) \\ s.t.\ y &= \mu_{x(0)},\ x \in [a,b]^n \end{aligned} \tag{4}$$

where the objective function is the solution $x = F(y)$ of the dynamical system in Equation (3) and the constraints are the initial fuzzy condition y described with a membership function $\mu_{x(0)}$ and the solution x being within the set $[a,b]^n$. In this problem, the optimal solution y^* is not important as $x_l(t^*) = F(y^*)$, which provides the lowest possible state value. The highest possible state value $x_h(t^*)$ is obtained by maximizing the same problem in Equation (4). The complete envelope of the dynamical system transient behavior, that is

the time evolution of the initial fuzzy state $\mu_{x(0)}$, is obtained by computing all the trajectories which lead to $x_l(t^*)$ and $x_h(t^*)$ for $t^* = 0 \ldots t_f$. The evaluation of the effect of process variation on the model trajectories is formulated similar to Equation (4). However, for a better convergence, the gradient of the objective function $x = F(p)$ with respect to the parameter p ($\frac{\partial x}{\partial p}$), which can numerically approximated using the relation ($\frac{\partial \dot{x}}{\partial p} = \frac{\partial \dot{x}}{\partial x}\frac{\partial x}{\partial p}$), has to be provided to the optimization engine [8]. In this work, we only used QS for input and initial conditions uncertainty types.

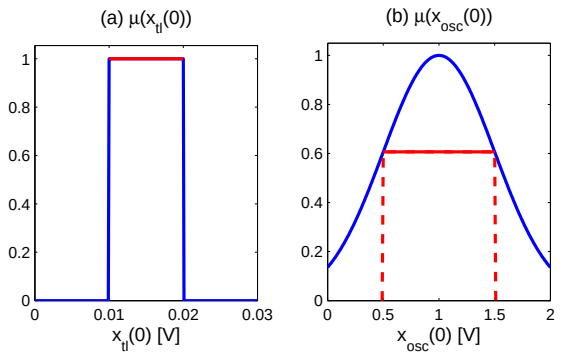

Fig. 1. Initial Conditions Membership Function Examples: (a) Uniform (b) Gaussian

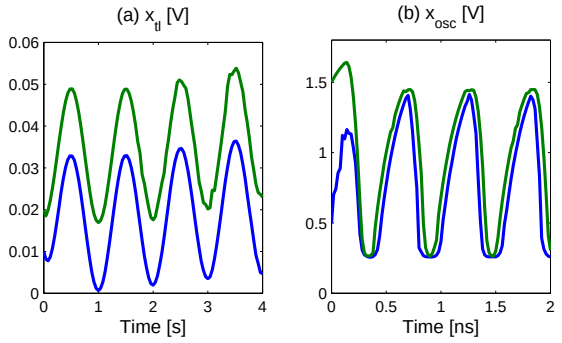

Fig. 2. Qualitative Simulation Examples : (a) Transmission Line Output (b) Differential Ring Oscillator Output

Figures 1(a) and 1(b) provide an example of fuzzy initial output conditions modeled with uniform and Gaussian membership functions for a transmission line and a differential ring oscillator, respectively. The QS of their output voltage is illustrated in Figures 2(a) and 2(b), respectively. The obtained transient envelopes contain a complete set of their model trajectories and provide an insight on how sensitive the model is to the considered fuzzy quantities. For example, Figure 2(a) shows that the transmission line behavior is equally affected while in Figure 2(b) the differential ring oscillator is more sensitive to its initial output state during the startup time.

IV. Proposed Methodology

Figure 3 presents the proposed method for reducing analog circuits mathematical models. It has five main components which are executed in an iterative way until the reduced model satisfies a defined set of accuracy and speedup constraints. First, in the model extraction and the pre-Analysis steps the circuit differential model is extracted and simulated in order to define a good guess of the MOR method parameters [5]. In this step, the model is linearized at different linearization points, which have been previously classified into regions via QS and clustering of the extended state space, in the previous pre-Analysis step. Then, the linearized models are reduced in each region using Krylov space projections and the final reduced model is obtained by dynamically evaluating a weighted sum of three local models. After that, the reduced model is input to a conformance criteria step to check that it is much faster than the original circuit model while it mimics its behavior. Based on the result of this step, either the reduced model is accepted or it is refined iteratively during the model refinement step until all the speedup and accuracy requirements are checked.

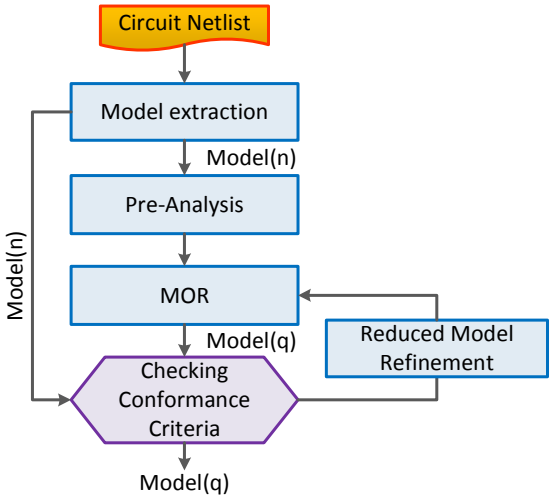

Fig. 3. MOR Method Overview

A. Model Extraction and Pre-Analysis

The model extraction step, which parses a circuit SPICE netlist and applies Modified Nodal Analysis (MNA) formulation [29], which leads to a parametric differential model ($Model(n)$), as given in Equation (1). It consists of n symbolic nonlinear differential equations relating all the state variables (the circuit voltages and currents) and including a set of parameters p which represent devices values and geometry. The main objective of the Pre-Analysis step is the evaluation of the full order $Model(n)$ and the prediction of a good initial guess of the MOR method parameters for a good reduced model accuracy/speedup tradeoff. In fact, $Model(n)$ is transformed to a set of FDEs, as given in Equation (3), and is input to QS. The input, initial conditions are considered as fuzzy numbers with uniform or Gaussian membership distributions, as detailed in Subsection III-C. The QS of the obtained FDEs leads to a set of state trajectories and their envelopes. The state trajectories are used to select linearization points and the envelopes provide a means to measure how sensitive the model is to each of the specified fuzzy quantities. If the state envelopes are tight, the model is not too sensitive to the introduced fuzziness. Otherwise, if the state envelopes are

wide, the model is very sensitive to the provided fuzzy quantity and the reduced model has to be built carefully in order to preserve the same behavior variations. As detailed in [5], the algorithm used for the selection of the k linearization points is based on k-means clustering. The linearization of Equation (1) is given in Equation (5).

$$\begin{aligned} \dot{x} &= A_{x_c} \cdot (x - x_c) + A_{u_c} \cdot (u - u_c) \\ &\quad + A_{p_0} \cdot (p - p_0) + f(x_c, t, u_c, p_0) \\ y &= g(x) \end{aligned} \tag{5}$$

where $A_{x_c} = \frac{df}{dx}_{|x=x_c}$, $A_{u_c} = \frac{df}{du}_{|u=u_c}$ and $A_{p_0} = \frac{df}{dp}_{|p=p_0}$.

One of the main shortcomings of previous MOR methods is the automatic selection of the linearization points set. In the proposed method, the extended state space which also includes the input, is considered and the linearization points selection is performed in two steps: (1) a subdivision of the circuit state space into R regions which are overlapping at their interface, where the behavior of the circuit is coarsely the same (extra-wide, wide, and tight QS envelope regions, etc.); and (2) the clustering via k-means of the content of each of these regions. The number of the linearization points, which ensure a target model accuracy in each of these regions, has to be minimal. The reason behind this is related to the speedup of the reduced model that is higher when using a smaller number of points. Figure 4 depicts this idea for a model having 2 state variables and requiring $R = 3$ behavior regions. The region R_1 contains many points, R_2 contains few points of R_1 and shares few points with R_3. Using this kind of regions subdivision, limits the total number of linearization points required to evaluate the reduced model during simulation which improves speedup and accuracy. For example, some points of regions 1 are required at the beginning of a simulation during a transient behavior and are never reused for the rest of the simulation time, while the remaining regions are alternatively used for a steady state behavior.

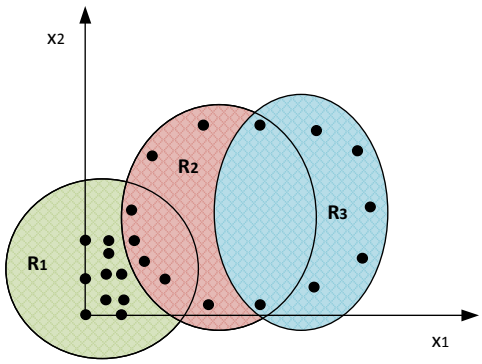

Fig. 4. State Space Regions

The objective of clustering is to gather similar states of the circuit in the same cluster and use their centroid (geometrical mean) as a linearization point. A local linear model around that point is going to be used to generate a local reduced model. However, using the centroid, which is not necessarily a real numerical solution of the circuit model, as a linearization point might introduce errors. Therefore, after clustering the content of each of the main R behavior regions using k-means clustering method, the clusters centroid are replaced with their nearest points from the simulation traces. The number of clusters k is not the same for all the R regions and is determined using the optimization procedure in Equation (6) below.

$$\begin{aligned} \min \quad & k \\ s.t. \quad & ||y - y_L|| \leq \varepsilon \end{aligned} \tag{6}$$

where the number of clusters k needs to guarantee a minimal error between the solution y of the original $Model(n)$, given in Equation (1) and the solution y_L of the piecewise linearized model given in Equation (5).

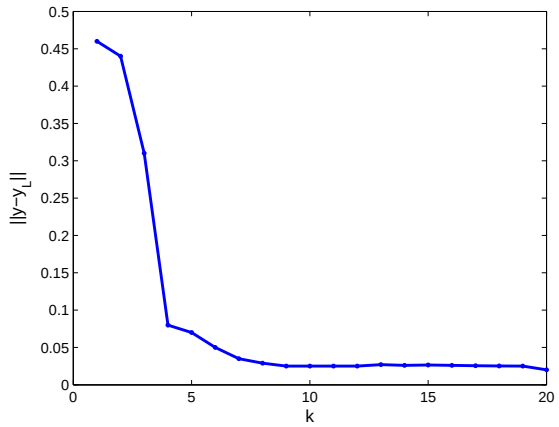

Fig. 5. Effect of the Number of Clusters on Model Accuracy

Figure 5 shows the error of a test circuit (Phase Locked-Loop) output for a varying number of clusters, using one behavior region after the lockup time. If we use less than 6 clusters the output error is greater than a predefined error threshold $\varepsilon = 0.05$. This error is going to increase after reducing the model. Then, we initially use a $k \geq 6$ as a first guess to generate a reduced model and we may tune it during the Model Refinement step, based on the Conformance Criteria Checking status.

B. Model Reduction

Figure 6 details the MOR method which consist of three main steps:

Step 1: Local Linear Models Generation: the matrices and the vectors in Equation (5) are computed for k_i clusters in each of the $i = 1 \ldots R$ regions.

Step 2: Reduction using Krylov Space Projection: a Krylov type projection basis V_i is determined for each of the $i = 1 \ldots R$ regions using the Arnoldi process [1]. It is a unified basis for the reduction via projection of all the local models within a region $V_i = SVD(\cup_{j=1}^{k_i} V_j)$, where SVD is the singular value decomposition operator [1]. The local reduced models $Model_i(q)$ are given in Equation (7), where z is the reduced state variable, i is the region index and j is the

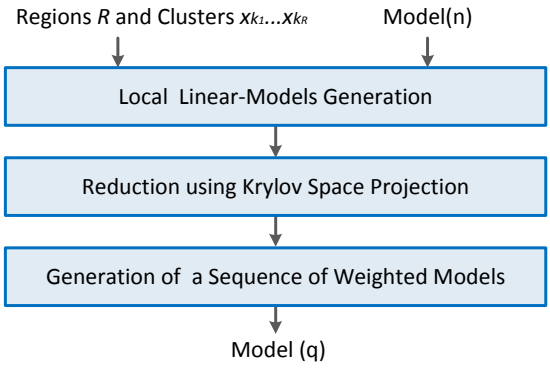

Fig. 6. Model Reduction Method

linearization point index. The terms related to the parameters are included only when the original model is sensitive to them. Otherwise, their presence in the reduced model will increase its evaluation time without necessarily improving its accuracy.

$$\begin{aligned} \dot{z} &= \hat{A}_{z_j} \cdot (z - z_j) + \hat{A}_{u_j} \cdot (u - u_j) \\ &\quad + \hat{A}_{p_0} \cdot (p - p_0) + \hat{f}(x_j, t, u_j, p_0) \end{aligned} \tag{7}$$

where $\hat{A}_{z_j} = V_i^t \cdot A_{x_j} \cdot V_i$, $\hat{A}_{u_j} = V_i^t \cdot A_{u_j}$, $\hat{A}_{p_0} = V_i^t \cdot A_{p_0}$, and $\hat{f}(x_j, t, u_j, p_0) = V_i^t \cdot f(x_j, t, u_j, p_0)$.

The matrices and the vectors in Equation (7) are stored in a lookup table and are used to compute on the fly the solution of the reduced differential model.

Step 3: Generation of a Sequence of Weighted Models: three reduced models $Model_i(q, p)$ are weighted to form the reduced model $Model(q)$. The weights are intended to smooth transitions between state space regions and allow contributions of different local models. However, the procedure for selecting the closest linearization points to the current state z and the weights computation should be simple, otherwise, the simulation time will increase extensively (more multiplication and summing operations) without any gain in terms of accuracy. We find the set of only three closest linearization points k_s such that $[k_s] = 3$ and it verifies the condition in Equation (8), where k_i represents the current region linearization points indexes.

$$\forall s \in k_s \ \forall j \in \{k_i - k_s\} \parallel \begin{matrix} z - z_s \\ u - u_s \end{matrix} \parallel \leq \parallel \begin{matrix} z - z_j \\ u - u_j \end{matrix} \parallel \tag{8}$$

We make sure that the selected closest points set k_s to the current state have been generated for similar input conditions by involving the input u as a constraint in Equation (8). Also, the fact that the linearization points are organized into sets, which correspond to different behavioral regions instead of a single region with many points, makes the search for the set of points k_s faster.

The final reduced model $Model(q)$ is given in Equation (9).

$$\begin{aligned} \dot{z} &= \sum_{s \in k_s} w_s \cdot \hat{A}_{z_s} \cdot (z - z_s) + \hat{A}_{u_s} \cdot (u - u_s) \\ &\quad + \hat{A}_{p_0} \cdot (p - p_0) + \hat{f}(x_s, t, u_s, p_0) \\ \hat{y} &= g(V_i \cdot z) \end{aligned} \tag{9}$$

where $\hat{y}$ is the output of the reduced model that approximates the output y of the full order model, $w_s = \frac{\|z - z_s\|^{-1}}{(\sum_{s \in k_s} \|z - z_s\|)^{-1}}, s \in k_s$ are the current state weights and the rest of the terms are as defined in Equation (7).

The main limitations of the above MOR method are the possible large number of state space regions and linearization points, when the circuit behavior is strongly nonlinear and the signal variability is high. Also, this method blindly reduces a flat circuit without optimizing the reduction effort when repeated circuit structures are present. These limitations can be avoided by subdividing a flat circuit into a set of sub-circuits. Therefore, this simple MOR method can be applied in an iterative way to reduce each sub-circuit in an order that depends on their computational cost, complexity and size, until a reduced model compliant with the conformance criteria is obtained, as described in the sequel.

C. Hierarchical MOR

The idea of hierarchical MOR of linear independent sub-circuits presented in [7], is extended here for the case of nonlinear circuit models based on the simple MOR shown in Figure 3. The advantage of a hierarchical reduction is to reduce the set of linearization points for large models and reduce the MOR computational effort when repeated circuit structures are present. For example, if a circuit has $N = 4$ transmission lines and each of them needs a set of $k = 7$ linearization points. The total number of linearization points required for the case of the simple MOR method is k^N $(7^4 = 2401)$ while this number is $N \times k$ $(7 \times 4 = 28)$ for the case of a hierarchical MOR method. Using a large number of linearization points slows down the reduced model considerably since during its evaluation the solution is always based on the closest points which are searched in the set of linearization points. The number of required transient regions R is also reduced in the same way.

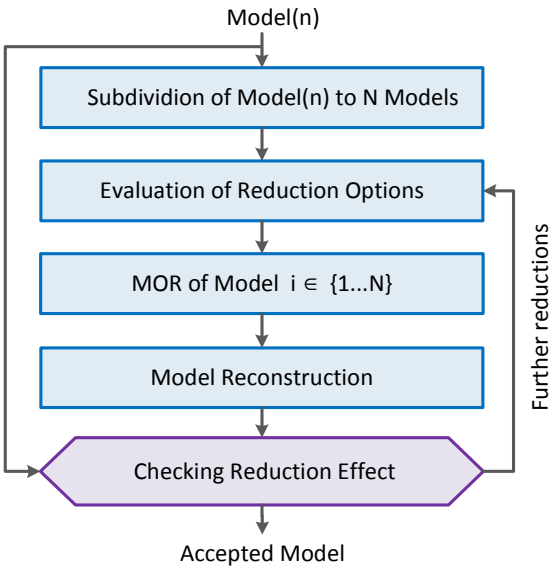

Fig. 7. Hierarchical MOR Scheme

Figure 7 provides the five main steps to perform a hierarchical MOR of large circuits having repeated sub-circuits by iteratively employing the simple MOR method, as follows:

$$\begin{array}{c} \\ eq1: \\ eq2: \\ eq3: \\ eq4: \\ \\ \\ \\ \\ \vdots \\ \\ eqn: \end{array} \begin{array}{c} \begin{array}{cccccccccccc} x_1 & x_2 & x_3 & x_4 & & & & & \dots & x_{n-1} & x_n \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & \dots & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 2 & 2 & 2 & & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 2 & & 0 & 0 \\ \vdots & & & \vdots & & & \vdots & & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & N & 0 \\ N & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & N \end{pmatrix} \end{array}$$

Fig. 8. Dependency Matrix

Step 1: Subdivision of $Model(n)$ into N models: if the circuit netlist is not defined as independent sub-circuits then the differential $Model(n)$ is subdivided into a set of N models (Equation (10)) while performing the MNA formulation.

$$\dot{x}_{Mi} \quad = \quad f_i(x_{Mi}, t, u, p) \tag{10}$$

where x_{Mi} is a subset of the state variables $x_1, \dots, x_n$.

The dependency matrix, whose rows refer to the $Model(n)$ equations indexes and columns refer to the state variables indexes, is generated as shown in Figure 8. Each element d_{ij} of this matrix is set to the number $i = 1 \dots N$ of the sub-circuit based on the dependency of the equation i on each of the state variable x_j, $j = 1 \dots n$.

Step 2: Evaluation of Reduction Options: the obtained N models are analyzed and the following quantities are computed:

- The complexity C quantified as the total number of nonlinear terms (nonlinear device) and the total number of linear terms (linear devices) divided by two (Equation (11)).

$$C = nbr(nonlinear\_terms) + \frac{nbr(linear\_terms)}{2} \tag{11}$$

- The percentage p_i of the total simulation time T_{tot} of all the N models (Equation (12)).

$$p_i = \frac{T_i}{T_{tot}} \times 100 \tag{12}$$

where T_i is the simulation time for the sub-circuit model i in Equation (10).

Based on this analysis, the sub-circuit model i which has the highest complexity and the largest percentage p_i of the total simulation time is reduced using the simple MOR method.

Step 3: Reduction of the model i: the simple MOR method, described in Figure 3, is applied to the sub-circuit model i leading to a speedup S_i that is computed as the simulation time of the original model i over its reduced model simulation time.

Step 4: Reconstruction of the circuit model: this is done using the reduced sub-circuit model i which leads to a partially reduced model. The expected speedup S_f of the reconstructed model is given in Equation (13) where S_i is the speedup of the reduced sub-circuit model i. It gives an idea of when the MOR in a hierarchical way can lead to a good overall speedup. In fact, if a part is consuming only $p_i = 30\%$ of the total simulation time, the speedup limit we can reach by reducing this model is $S_{lim} \approx 1.4$. However, if a part is worth $p_i = 90\%$ of the total computational time, the speedup limit that can be reached by reducing this model is $S_{lim} \approx 10$. This emphasizes that the hierarchical MOR method is more effective for large circuits with many repeated similar entities.

$$S_f = \frac{1}{1 + \frac{p_i}{100}\ (\frac{1}{S_i} - 1)} \tag{13}$$

Step 5: Reduction Effect: the reconstructed model speedup and accuracy are checked by simulation. Based on the result of this step, either the reconstructed model is accepted or additional reduction effort is performed on a new candidate from the remaining sub-circuits models.

D. Checking Conformance Criteria

The objective of this step is to check that the reduced model fulfills some conformance criteria (accuracy and speedup) and can effectively be used instead of the original model for system level simulations. Because of its piecewise nature, the reduced models cannot be as accurate as the original nonlinear model for all inputs and conditions ranges. However, the proposed method strengthens the accuracy of the generated models by construction and through the use of qualitative simulation that defines the performance bounds of the model. For example, if the original model has a strong nonlinear behavior related to the input variation, the reduced model is compared to the original model for complete ranges of inputs. The system in Equation (14) below provides a few examples of conformance criteria used to verify that the reduced model is behaviorally equivalent to the original model, where the error tolerances ε_1, ε_2, ε_3, and ε_4 depend on the level of accuracy of the application that will make use of the reduced model. In practice, relative error values are used to measure the accuracy of two curves and are considered probably acceptable and meaningful in the range $3-5\%$, good if less than 2%, and excellent if less than 1% [30].

$$\begin{aligned} \frac{\|\hat{x} - x\|_2}{\|x\|_2} &\leq \varepsilon_1 \\ \frac{\|\hat{y} - y\|_2}{\|y\|_2} &\leq \varepsilon_2 \\ \|Freq(y) - Freq(\hat{y})\| &\leq \varepsilon_3 \\ \|DC(y) - DC(\hat{y})\| &\leq \varepsilon_4 \\ S(Model(q)) = \frac{T_{sim}(Model(n))}{T_{sim}(Model(q))} &\geq S_{min} \end{aligned} \tag{14}$$

In order to check that the input-output relationship of the original model is preserved, the relative error threshold of the output is required to be ($\varepsilon_2 \leq 2\%$). The relative error

threshold of the state variables, which are expected to have some deviations from the original behavior without affecting the output, is required to be ($\varepsilon_1 \leq 3-5\%$) or better. The frequency of the original model and the frequency of reduced model outputs have to be the same ($\varepsilon_3 \leq 2\%$). The DC characteristics of the original and reduced models are required to be accurate ($\varepsilon_4 \leq 2\%$) in addition to preserving the same properties (stability, hysteresis, gain, noise margin, etc.) [31]. The reduced model has to achieve a minimum speedup compared to the original model. Also, the conformance criteria are evaluated for a range of inputs and initial conditions. If the conformance criteria are not satisfied, a refinement of the MOR parameters step is conducted. During this step, the conformance criteria which are not met are investigated and the MOR parameters are iteratively adjusted until the requirements are satisfied which leads to an acceptance of the reduced model $Model(q)$.

E. Reduced Model Refinement

The reduced model refinement step consists in tuning the parameters of the MOR method which affects the accuracy and speedup of the reduced model, based on the conformance criteria checking result. The simulation traces of the reduced model are analyzed and the regions, where the conformance criteria are not met, are determined and used to update the parameters which can lead to a better speedup and accuracy tradeoff.

Algorithm 1 Reduced Model Refinement Example

1: **Input:** S_{min}, $S(Model(q))$, ε_1, ε_2, $K = \{k_1, k_2, \ldots, k_R\}$, k_{max},R_{max}, V, x, z, y, $\hat{y}$, q, q_{min}
2: **Output:** $K_{up} = \{k_1, k_2, \ldots, k_{R_{up}}\}$, R_{up}, V_{up}, q_{up}
3: **if** $Error(x)$ **or** $Error(y)$ **then**
4: $i \leftarrow$ Find$(K, x, \hat{x}, y, \hat{y}, \varepsilon_1, \varepsilon_2)$
5: **if** $k_i \leq k_{max}$ **then**
6: $k_i \leftarrow$ Increase(k_i)
7: **else if** $R < R_{max}$ **then**
8: $R_{up} \leftarrow$ Increase(R, k_i)
9: **else**
10: Return("Accuracy can not be improved!")
11: **end if**
12: $K_{up} \leftarrow$ Verify(Equation (6))
13: **else if** $S(Model(q)) \leq S_{min}$ **then**
14: **if** $\frac{S(Model(q))}{S_{min}} > 0.75$ **and** $R < R_{max}$ **then**
15: $R_{up} \leftarrow$ Increase(R, K)
16: $K_{up} \leftarrow$ Verify(Equation (6))
17: **else if** $q > q_{min}$ **then**
18: $q_{up} \leftarrow$ Decrease(q)
19: **else**
20: Return("Speedup can not be improved!")
21: **end if**
22: **end if**
23: **for** $i = 1 \ldots R_{up}$ **do**
24: $V_i \leftarrow SVD(\cup_{j=1}^{k_i} V_j)$
25: **end for**
26: $V_{up} \leftarrow \{V_1, V_2, \ldots, V_{R_{up}}\}$

Algorithm 1 is a simplified form of the reduced model refinement process for the simple MOR method. It requires as inputs, in line 1, the conformance checking criteria, the simulation traces of $Model(n)$ and $Model(q)$ and the current MOR parameters and their maximal and minimal values. It outputs, in line 2, the number of regions R, the number of clusters k_i in each region, the unified projection basis for each region, and the reduction order q which is also the size of the reduced model. The accuracy criteria of the reduced model is addressed before the speedup criteria in a recursive way. In lines $3-9$, the number of linearization points k_i is increased in order to fix the behavior of the reduced model in the region R_i (determined in line 4), which does not meet the accuracy conformance criteria. If the number of the linearization points k_i reach the limit k_{max}, the set of regions is increased by splitting the region R_i into two new regions. After updating the number of clusters or the number of regions, the piecewise linearized model accuracy is verified in line 12 using Equation (6). In lines $13-16$, the number of regions R is increased first if it is less than the limit R_{max} and the speedup value is within the $75-100\%$ of the target speedup S_{min} and Equation (6) is verified. Otherwise, the size of the reduced model q is decreased to improve its speedup $S(Model(q))$ in lines $17-18$. In all refinement cases, the unified reduction basis V_i in all regions are updated in lines $23-25$, since the piecewise linear approximations are changed according to the refinement process. The reduced model refinement process is performed independently for each of the failing regions. Because of that, it is usually performed at the cost of several simulations of the reduced model for the failing regions until the required accuracy constraint is met. In lines 10 and 20, the algorithm outputs that no further refinements can be done to check the current conformance criteria. In this case, the HMOR might become a good alternative to the simple MOR method.

V. Applications

In this section, the simple MOR method is applied to a nonlinear transmission line, a ring oscillator, a voltage controlled oscillator, a phase locked loop and an analog comparator. Also, we perform a PLL model reduction using the hierarchical MOR presented in Subsection IV-C. All simulations were performed in the MATLAB environment [11], on a $Windows\ 7$ operating system with an Intel core $I7$ CPU, $2.8GHz$ with $24GB$ of RAM. In the following applications, we refer to the size of the original model and the reduced model as $Model(n)$ and $Model(q)$ where n is the size of the original model, q is the size of the reduced model and $n > q$.

A. Transmission Line

Figure 9 shows the transmission line model that is a chain of connected resistor, capacitor and nonlinear diode cells. The input current source is $i(t)$ and all capacitors and resistors values are set to $1\ F$ and $1\ \Omega$, respectively. The behavior of the diodes is nonlinear and is given by $I_d(v) = \exp(40v) + v - 1$.

The full order model of the transmission line is given in Equation (15), where $x_1, \ldots, x_n$ are the circuit node voltages.

$$\begin{aligned} \dot{x_1} &= -I_d(x_1) - I_d(x_1 - x_2) + b\ i(t) \\ \dot{x_i} &= I_d(x_{i-1} - x_i) - I_d(x_i - x_{i+1}) \\ \dot{x_n} &= I_d(x_{n-1} - x_n) \\ y &= x_1 \end{aligned} \tag{15}$$

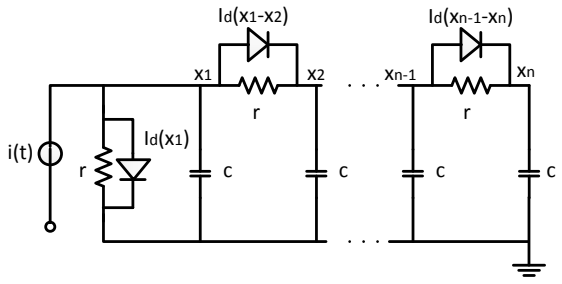

Fig. 9. Transmission line with nonlinear diodes

In this application, we used 3 main transient regions and $k = 20$ linearization points. We also considered three cases of inputs and problem sizes as follows:

- Case I: $i(t) = H(t-3)$, $n = 1500, q = 30$
- Case II: $i(t) = exp(-t)$, $n = 1500, q = 30$
- Case III: $i(t) = sin(\frac{2\pi t}{10})$, $n = 100, q = 10$

Figure 10 shows that the transient behavior of the full order model and the reduced order model problems of Table I is the same, for the above three simulation cases.

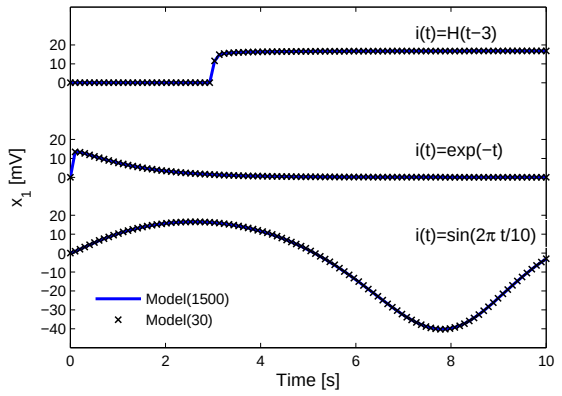

Fig. 10. Transmission line transient behavior

Table I compares the simulation times of the TPWL [3], PWP [4] and our method for the transmission line. Although the shown results were conducted using different processors, the speedup criteria can measure the improvement of the proposed MOR method. In fact, the generated reduced models are more than 10 times faster than the TPWL and our plain implementation of the PWP method and 1000 times faster than the full nonlinear model for the first two large scale problem sizes.

TABLE I
SIMULATION TIMES AND SPEEDUP FOR THE NONLINEAR TRANSMISSION LINE CIRCUIT USING $k = 20$ LINEARIZATION POINTS

Case	TPWL [3]	PWP [4]	Proposed Method
I	$\frac{9573.30}{80.80} \simeq 118$	$\frac{873.38}{17.16} \simeq 51$	$\frac{810.39}{0.64} \simeq 1248$
II	$\frac{11713.10}{110.90} \simeq 105$	$\frac{986.32}{43.53} \simeq 23$	$\frac{1061.32}{0.82} \simeq 1284$
III	$\frac{25.40}{2.70} \simeq 9$	$\frac{4.6}{2.4} \simeq 2$	$\frac{1.84}{0.31} \simeq 6$

Table II shows that the proposed method reduced models mimic the behavior of their original models for the three considered problem cases. The accuracy criteria is satisfied for the different experiments and the relative errors of the state variables and the output are always less than 10^{-2}, that is the maximum acceptable error during the conformance criteria checking step. Unexpectedly, the PWP method, which employs local second order Taylor polynomial approximations, is less accurate than the TPWL and our method, which employ only local linear approximations, for the first two problem cases. This can be explained by the fact that the aggregation of multiple projection basis for the first order terms and second order terms in the PWP method can sometimes lead to large errors. An additional drawback of the PWP is a large memory requirement to store the second order matrices. For example, the size of the projection matrices of the second order terms for the cases I and II is $size(V \otimes V) = (1500 \times 30)^2$ which requires a minimum memory of $(1500 \times 30)^2 \times 8\ Bytes = 16.2\ GB$ assuming a double precision number of 8 Bytes. Therefore this method is impractical and very computationally expensive for large models.

TABLE II
ACCURACY FOR THE TRANSMISSION LINE CIRCUIT USING $k = 20$ LINEARIZATION POINTS

Case	TPWL [3]	PWP [4]	Proposed Method
I	$0.3\ 10^{-2}$	$11.69\ 10^{-2}$	$0.12\ 10^{-2}$
II	$0.48\ 10^{-2}$	$4.27\ 10^{-2}$	$0.18\ 10^{-2}$
III	$0.56\ 10^{-2}$	$0.56\ 10^{-2}$	$0.37\ 10^{-2}$

B. Ring Oscillator

Figure 11 represents a ring oscillator composed of a large odd number n of inverters connected in a circular chain. Each inverter is single ended and is composed of a cascaded $nmos$ and $pmos$ transistors and a capacitance C connected to their drains. The node voltages x_i of each of the n inverter oscillates between the ground $gnd = 0V$ and the power $vdd = 1.8V$.

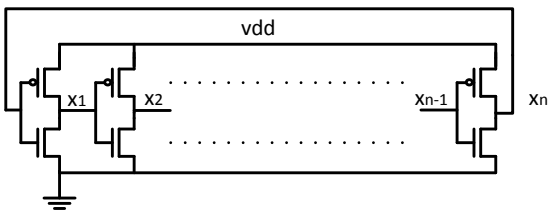

Fig. 11. Ring oscillator circuit

The circuit model is given in Equation (16), where $x_i,\ i = 1 \ldots n$, are the node voltages, $C = 0.164fF$ and the functions I_n and I_p model the nonlinear current generated by the $nmos$ and $pmos$ transistors, respectively, based on their gate, drain and source voltages. The initial conditions $x(0)$ are represented by the fuzzy number $\mu_{x(0)}$.

$$\begin{aligned} \dot{x_1} &= -\frac{1}{C}(I_n(x_n, x_1, gnd) + I_p(x_n, x_1, vdd)) \quad (16) \\ \dot{x_i} &= -\frac{1}{C}(I_n(x_{i-1}, x_i, gnd) + I_p(x_{i-1}, x_i, vdd)) \\ y &= x_n \\ x(0) &= \mu_{x(0)} \end{aligned}$$

When using one transient region for the ring oscillator model, it is hard to reproduce the oscillation behavior of the full order $Model(131)$ with the required speedup and accuracy constraints, ($S_{min} = 100$) and ($\frac{\|\hat{x}-x\|_2}{\|x\|_2} \leq 10^{-2}$ and $\frac{\|\hat{y}-y\|_2}{\|y\|_2} \leq 10^{-2}$), respectively. This could be explained by the highly nonlinear initial startup transient region that needs to be accurately approximated by the reduced model. This highlights two key points of our method, namely the need for a further subdivision of the transient behavior into smaller regions, and the QS that provides better coverage of the initial region.

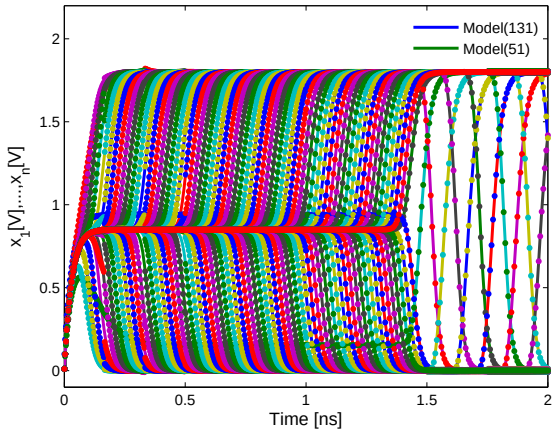

Fig. 12. Ring oscillator internal state transient responses

Our method used 10 transient regions to provide a reduced model that mimics accurately the full order model behavior as shown in Figure 12, where the state vector of the original full order model x is represented by the solid line and the backward projection of the reduced order model $\hat{x} = Vz$ is represented by the dotted line.

Table III presents the refinement of the MOR for the ring oscillator model. The speedup and accuracy requirements were satisfied after refinement of the number of clusters in each region with $S_{Ring} = 139$ and $\frac{\|\hat{x}-x\|_2}{\|x\|_2} = 0.4\ 10^{-2}$ and $\frac{\|\hat{y}-y\|_2}{\|y\|_2} = 0.63\ 10^{-2}$.

TABLE III
REFINEMENT OF $Model(51)$ FOR THE RING OSCILLATOR MODEL

Number of clusters k	Speedup S_{Ring}	$\frac{\|\hat{x}-x\|_2}{\|x\|_2}$	$\frac{\|\hat{y}-y\|_2}{\|y\|_2}$	Status
10	$\frac{142.33}{0.68} \simeq 207$	$0.99\ 10^{-2}$	$2.33\ 10^{-2}$	rejected
14	$\frac{142.33}{0.71} \simeq 199$	$0.55\ 10^{-2}$	$1.30\ 10^{-2}$	rejected
18	$\frac{142.33}{0.96} \simeq 147$	$0.43\ 10^{-2}$	$1.76\ 10^{-2}$	rejected
19	$\frac{142.33}{1.02} \simeq 139$	$0.40\ 10^{-2}$	$0.63\ 10^{-2}$	accepted

C. Voltage Controlled Oscillator

We apply the MOR method to the current-starved Voltage Controlled Oscillator (VCO) detailed in [32]. Figure 13 shows the schematic of the VCO that consists in two main components: an inverter chain composed of the $pmos$ and $nmos$ transistors at the center and a current mirror structure (upper $pmos$ and lower $nmos$ transistors) that limits the current mirrored in each of the inverters.

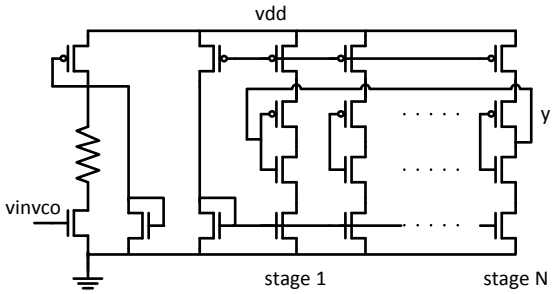

Fig. 13. VCO circuit

The QS of the VCO output voltage using a fuzzy input μ_{vinvco} returns that it is very sensitive to the input voltage $vinvco$ which determines its oscillating frequency, as shown in Figure 14. Using the full order VCO simulation, we determined 50 clusters within four main regions to build the reduced model. The obtained reduced VCO model has 35 state variables and is 23 times faster than the original VCO model ($S_{VCO} = \frac{8708.73s}{378.61s} = 23$).

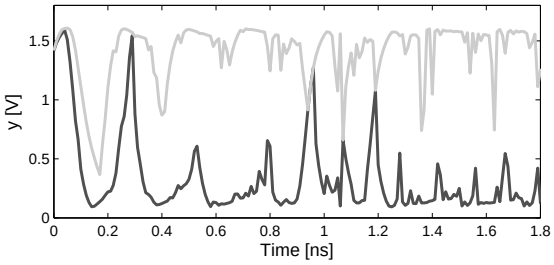

Fig. 14. QS of the VCO output

Figure 15 compares the frequencies of the VCO full order $Model(48)$ and the reduced $Model(35)$ output signals, obtained using Fast Fourier Transform [11]. It shows that both models are oscillating at the exact frequencies for the specified input voltage range.

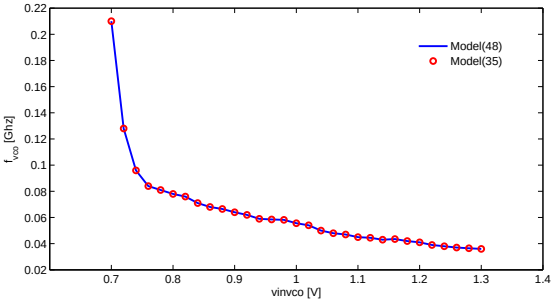

Fig. 15. Comparison of the VCO output frequency

Figure 16 compares the state variables and the output of the full order $Model(48)$ and the backward projection of the reduced $Model(35)$. It proves that accuracy conformance criteria are verified by having $\frac{\|\hat{x}-x\|_2}{\|x\|_2} \leq 0.02$ and $\frac{\|\hat{y}-y\|_2}{\|y\|_2} \leq 0.02$.

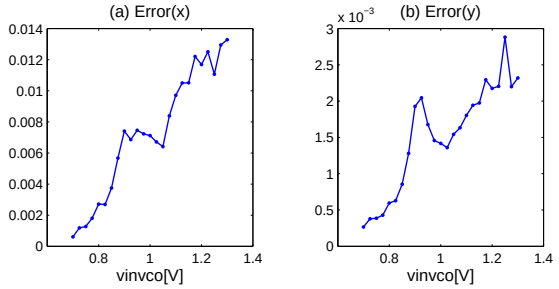

Fig. 16. Accuracy of the VCO reduced model

TABLE IV
COMPARISON OF THE EFFECT OF QS, CLUSTERING, REGIONS SUBDIVISION, AND WEIGHTING LOCAL MODELS

Without	QS	Clustering	Regions	Weights
$\frac{\|\hat{x}-x\|_2}{\|x\|_2}$ $[10^{-2}]$	42.1	12.10	3.37	2.38
$\frac{\|\hat{y}-y\|_2}{\|y\|_2}$ $[10^{-2}]$	19.08	9.74	1.89	1.71
S_{VCO}	24	5	11	27

Table IV compares the effect of QS, clustering, regions subdivisions and the use of the weighted closest local models. In each column one component of the presented method is omitted and the obtained reduced VCO model is evaluated. When the QS is not used and is replaced with training the original VCO model in column 2, a slight perturbation of the input leads to large relative error values. This is due to the fact that important trajectories, which correspond to the input variation, were not considered and because of that the linearized model was not accurate. Using a linearization points selection procedure based on traversing the training trajectories and comparing the linear approximation with the original VCO trajectory instead of clustering in column 3 leads to inaccurate approximation and a smaller speedup value. This is due to the larger number of linearization points and a resulting lossy projection matrix. The effect of omitting the regions subdivision leads also to higher error values and smaller speedup value. This is expected since a single projection matrix is used instead of four different projection matrices for each of the separate four behavioral regions. The last column 5 shows that the obtained reduced VCO model is fast but is not as accurate as the refined VCO model where all these techniques are employed as shown in Figure 16.

D. Phase-Locked Loops

We consider a $cmos - 180nm$ implementation of a Phase-Locked Loop (PLL), as described in [32]. This circuit is frequently used in the front-end of modern integrated circuits. It is a nonlinear frequency-control system that generates a clock signal that locks after a delay time (the locking time) given an input data signal. The PLL is said to be locked when the input and the feedback clock frequencies match. However, the PLL might not lock for various reasons such as an input frequency out of the specified range, jitter, noise, reset situations, or when any of the PLL sub-circuits is not behaving properly [33].

Figure 17 shows a simplified block diagram of the major sub-circuits of the PLL of this application. This PLL is modeled at the transistor level with a set of 62 differential equations and has the following main blocks:

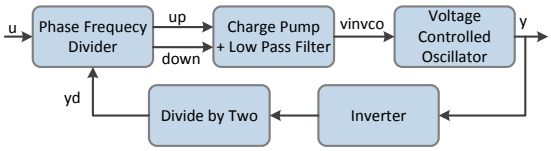

Fig. 17. PLL Block Description

- A Phase Frequency Detector (PFD), that detects the difference in phase and frequency between the input clock u and feedback clock y_d, and asserts an up or $down$ control signal based on whether the feedback signal y_d frequency is lagging or leading the input u frequency.

- A Charge Pump (CP) and a Low Pass Filter (LPF), that receives the up and $down$ signals from the PFD and drives a current to the LPF if the up signal is high and draws a current from the LPF if the $down$ signal is high.

- A Voltage Controlled Oscillator (VCO), which is biased with the control voltage $vinvco$ generated by the LPF sub-circuit. It oscillates at a higher frequency if the $vinvco$ signal increases and oscillates at a lower frequency, otherwise. Consequently, the oscillation frequency of the VCO affects directly the phase and frequency of the feedback clock signal y. It stabilizes at a fixed frequency when the output of the LPF settles to a DC voltage.

- An inverter, which is a buffer inverting the output of the VCO.

- A feedback divide by two sub-circuit that increases the frequency of the VCO output signal generating a signal y_d that is input to the PFD.

The full order PLL $Model(62)$ was reduced to $Model(7)$ using 21 regions and a number of 10 clusters in each of these regions. The first 18 regions were required to approximate the start up behavior and after the locking time, only 3 regions are required to reproduce the behavior of the original PLL.

Figure 18 shows the verification of the locking property for two PLLs: the original PLL (on the bottom) and the reduced PLL (on the top). The VCO input signal reaches a stable value which make both PLLs lock almost at the same time $T_{lock} = 0.3\mu s$.

Figure 19 shows three internal voltages of the full order PLL $Model(62)$ on the left side and the reduced PLL $Model(7)$ on the right side. The slight deviation of the signals is expected because the MOR process is lossy. However, this deviation does not affect the VCO output that still oscillates at the expected frequency.

Table V compares the PLL internal voltages with their respective approximations generated with the PLL reduced model for different input clock frequencies. Basically, it shows that the reduction is quite accurate. In all cases, the reduced PLL of size $Model(7)$ is more than 40 times faster than the full order model $Model(62)$.

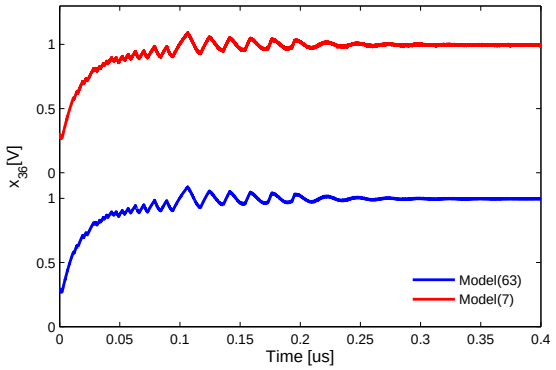

Fig. 18. PLL locking signal comparison

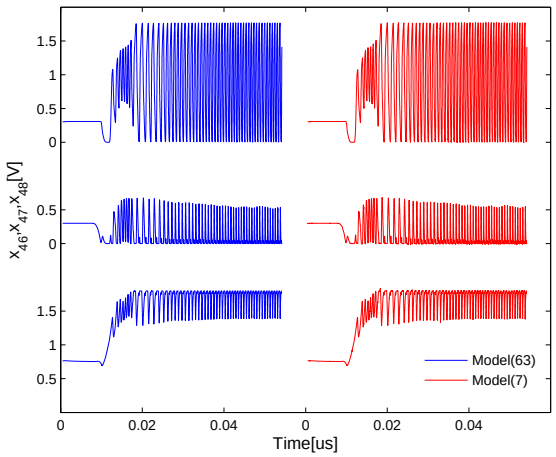

Fig. 19. PLL internal voltages comparison

TABLE V
REDUCED PLL PERFORMANCE FOR DIFFERENT INPUT FREQUENCIES

Frequency$[Ghz]$	$\frac{\|\hat{x}-x\|_2}{\|x\|_2}$	$\frac{\|\hat{y}-y\|_2}{\|y\|_2}$	Speedup S_{PLL}
1	$1.87\ 10^{-2}$	$0.38\ 10^{-2}$	41
0.8	$0.56\ 10^{-2}$	$0.46\ 10^{-2}$	43
0.6	$1.03\ 10^{-2}$	$0.23\ 10^{-2}$	43

E. PLL Reduction using Hierarchical MOR

Following the procedure described in Subsection IV-C, we performed a subdivision of a PLL original model, having 813 state variables, into 4 sub-circuits and analyzed their performance, as shown in Table VI. The VCO (255 stages) and the inverter, which are modeled with 768 nonlinear equations, are the most computationally expensive part that uses 96.85% of the total PLL simulation time. These statistics make the VCO and the inverter sub-circuit the ultimate candidate for the HMOR reduction type of the PLL. Based on the formula in Equation (13), we can target at least a PLL simulation speedup of $S_{PLL} = \frac{1}{1+\frac{96.85}{100.}\ (\frac{1}{1000}-1)} \approx 30$, if the VCO and the inverter reduced sub-circuit has a speedup $S_{VCO+Inverter} \geq 1000$.

Table VII compares the accuracy and speedup results when reducing the PLL as a flat circuit using the simple MOR method or reducing the VCO and the inverter sub-circuit using the HMOR method. The original PLL $Model(813)$ is reduced to $Model(105)$ and $Model(75)$ using the same number of linearization points and the same regions subdivision, in the first two rows and the last two rows, respectively. The reduced PLL models obtained via the HMOR method are more accurate and slightly faster than the ones via the simple MOR method. The better accuracy results when using the HMOR method are expected because the unreduced PLL sub-circuits (PFD, CP, LPF, and Divide by Two) are modeled with their original nonlinear equations while only the VCO and the inverter, which have the same uniform structure, are reduced. The speedup of the reduced sub-circuit is $S_{VCO+Inverter} \geq 1300$ and the obtained PLL speedup is as expected in both HMOR cases. The slightly smaller speedup for the simple MOR method is explained by the fact that replacing a small number of nonlinear equations for the sub-circuits (PFD, CP, LPF, and Divide by Two) with a piecewise linear representation is not expected to lead to higher speedup values. In fact, the speedup is considerable only when the reduction ratio is important because the matrices involved in a reduced model are dense and the number of multiplication and summations becomes comparable to the nonlinear model case for small reduction ratios.

TABLE VI
PLL SUB-CIRCUITS ANALYSIS REPORT

sub-circuits	Percentage(%) Simulation time p_i	N° of Equations	Complexity C
PFD	1.92	30	60
CP and LPF	0.71	9	14.5
VCO and Inverter	96.85	768	1027.5
Divide by Two	0.65	6	9

TABLE VII
COMPARISON OF PLL SIMULATION RESULTS

PLL Reduction	Size q	$\frac{\|\hat{x}-x\|_2}{\|x\|_2}$	$\frac{\|\hat{y}-y\|_2}{\|y\|_2}$	Speedup S_{PLL}
HMOR	$60+45$	$0.35\ 10^{-2}$	$4.49\ 10^{-3}$	31.36
MOR	105	$1.03\ 10^{-2}$	$1.25\ 10^{-2}$	27.10
HMOR	$30+45$	$0.81\ 10^{-2}$	$2.49\ 10^{-3}$	31.56
MOR	75	$3.53\ 10^{-2}$	$2.89\ 10^{-2}$	30.93

F. Analog Comparator

We consider a $cmos-180nm$ analog comparator circuit, as described in [32]. It is a decision-making circuit composed of three main stages: a pre-amplification (a differential amplifier with active loads), a decision circuit (a positive feedback) and a post-amplification (a self biasing differential amplifier used as a buffer), as shown in Figure 20. If the positive input voltage v_p is greater than the negative input voltage v_m, the comparator output is set to the maximum voltage $v_{out} = 1.8V$. Otherwise, the comparator output is set to the minimum voltage $v_{out} = 0V$. In practice, the propagation delay, the sensitivity and the noise rejection of the comparator are of a great concern.

Using a SPICE comparator netlist, we elaborated a differential model $Model(16)$ having 16 state variables using a set of nominal parameters p_0 representing the width and length of each of the 19 comparator transistors. $Model(16)$ has been

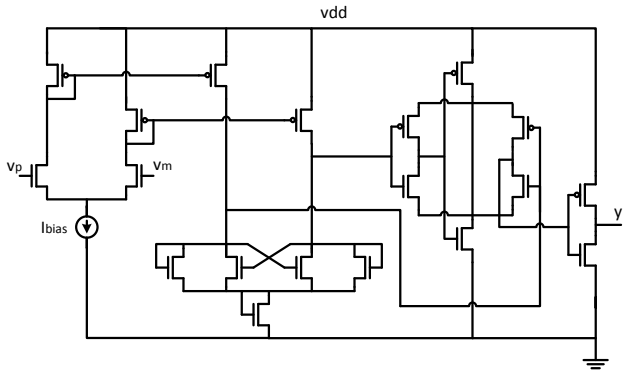

Fig. 20. Analog comparator circuit

reduced using 10 behavioral regions to a weighted piecewise linear model $Model(5)$ having only 5 reduced state variables, as given in Equation (17).

$$\begin{aligned} \dot{z} &= \hat{A}_{z_j}(z - z_j) + \hat{A}_{v_{pj}}(v_p - v_{pj}) \\ &\quad + \hat{A}_{v_{mj}}(v_m - v_{mj}) + \hat{f}(x_j, t, v_{pj}, v_{mj}, p_0) \\ \hat{x} &= V_i^t \; z \\ v_{out} &= \hat{x}(16) \end{aligned} \tag{17}$$

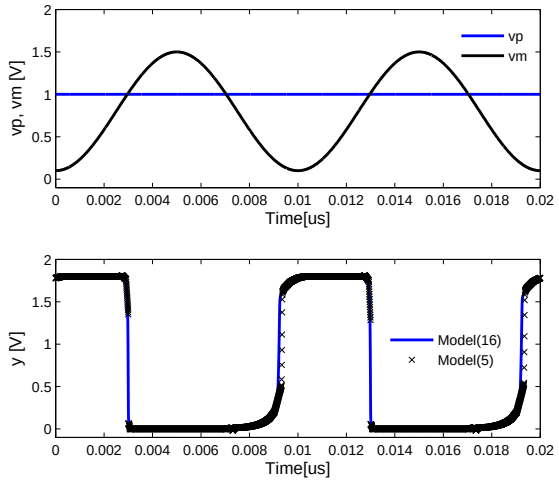

Fig. 21. Comparator inputs and corresponding output for sine input v_m

Figure 21 compares the behavior of the comparator reduced model $Model(5)$ to the full order model while the input is such that the input v_p is held to $1V$ and the input $v_m = 0.8 - 0.7\cos(2\pi \times 10^8)$. The output voltage v_{out} is accurately set to its maximum value when $v_p \geq v_m$.

Figure 22 shows the case when the input v_m is held to $0.9V$ and the input v_p is swept up and down between $0V$ and $1.8V$. This experiment shows that the reduced model has the same DC characteristic, the same offset voltage and the same hysteresis behavior that prevents its output instability.

Figure 23 shows the case when the input v_m is held to $1.2V$ and the input v_p is a pulse of a $10ns$ duration and reaching a maximum voltage of $1.25V$. This experiment proves that the comparator reduced model has the same sensitivity of $50mv$.

Finally, Table VIII provides accuracy and speedup results for the reduced comparator $Model(5)$ for different inputs. The

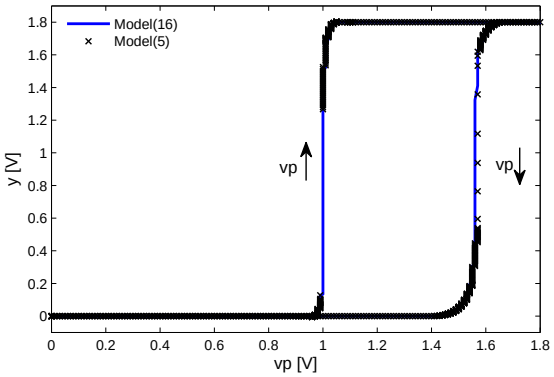

Fig. 22. Comparator inputs and corresponding output for DC sweep input

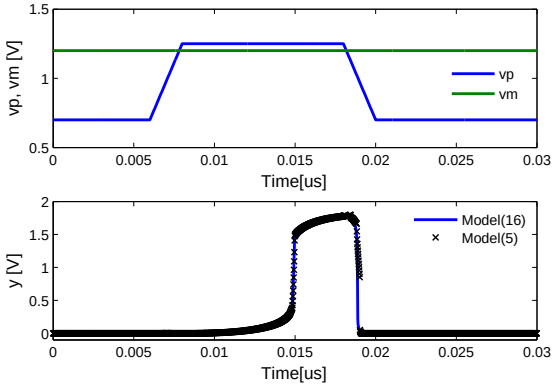

Fig. 23. Comparator inputs and corresponding output for pulse input

simulation results of rows 1,2 and 3 are shown respectively in Figures 21, 22, and 23.

TABLE VIII
PERFORMANCE OF THE COMPARATOR $Model(5)$ FOR DIFFERENT INPUTS

Input Case	$\frac{\|\hat{x}-x\|_2}{\|x\|_2}$	$\frac{\|\hat{y}-y\|_2}{\|y\|_2}$	Speedup S_{comp}
Figure 21	0.044	0.107	$\frac{132.44}{12.06} = 11$
Figure 22	0.045	0.063	$\frac{194.44}{21.57} = 9$
Figure 23	0.014	0.107	$\frac{171.34}{13.46} = 13$

VI. CONCLUSION

This paper presented a methodology for model order reduction of nonlinear analog circuits. Different techniques such as qualitative simulation, k-means clustering, linearization, and Krylov projections were used to build reduced models which are more robust to small signal and parameters variations. The selection of an initial guess of the number of linearization points was addressed as an optimization problem minimizing the behavior error between the full order model and its full order linearized form. The use of the input as a decision variable in an extended state space enhanced the method and improved its accuracy. A hierarchical MOR method that subdivides a circuit model into different blocks was also added as an option to meet specific accuracy and reduction requirements

and optimize the reduction effort. The experimental results on several analog circuits show that the presented method is accurate and effective. The comparison of our results with two main related methods [3] [4] shows better speedup and accuracy results for the transmission line model. However, our experiments with these methods for the remaining applications show that they fail to preserve their behaviors (large relative errors $> 10\%$). In future work, the reduced model refinement process can be extended to generate the specifications and limitations of the reduced models automatically. Also, a statistical or a learning based method, which selects a reduced model from a set different reduced models given a target performance requirements, can be implemented. Furthermore, the implementation of a technique that handles the interface between adjacent behavioral regions can enhance the accuracy of the reduced models. Finally, the developed framework can be used to enhance statistical simulation and circuit synthesis.

REFERENCES

[1] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*. Society for Industrial and Applied Mathematic, 2005.

[2] S. Tab and L. He, *Advanced Model Order Reduction Techniques in VLSI Design*. New York, USA: Cambridge University Press, 2007.

[3] M. Rewienski and J. White, "A Trajectory Piecewise-Linear Approach to Model Order Reduction and Fast Simulation of Nonlinear Circuits and Micromachined Devices," *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 22, no. 2, pp. 155–170, 2006.

[4] N. Dong and J. Roychowdhury, "General-Purpose Nonlinear Model-Order Reduction Using Piecewise-Polynomial Representations," *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 27, no. 2, pp. 649–654, 2008.

[5] H. Aridhi, M. H. Zaki, and S. Tahar, "Towards Improving Simulation of Analog Circuits Using Model Order Reduction," in *ACM/IEEE Design, Automation and Test in Europe*, pp. 1337–1342, 2012.

[6] M. Hinze, M. Kunkel, and M. Vierling, "POD model Order Reduction of Drift Diffusion Equations in Electrical Networks," in *Model Reduction for Circuit Simulation*, vol. 74 of *Lecture Notes in Electrical Engineering*, pp. 177–192, Springer, 2011.

[7] M. Honkala, P. Miettinen, J. Roos, and C. Neff, "Hierarchical Model-Order Reduction Flow," in *Scientific Computing in Electrical Engineering*, Mathematics in Industry, pp. 539–546, Springer Berlin Heidelberg, 2010.

[8] G. Bontempi, "Simulating Continuous Dynamical Systems Under Conditions of Uncertainty: The Probability and The Possibility Approaches," in *Fuzzy Partial Differential Equations and Relational Equations*, vol. 142 of *Reservoir Characterization and Modeling Series: Studies In Fuzziness and Soft Computing*, pp. 130–152, Springer, 2004.

[9] C. Jacoboni and P. Lugli, *The Monte Carlo Method for Semiconductor Device Simulation*. Springer Vienna, 1989.

[10] R. W. Freund, "Krylov-Subspace Methods for Reduced-Order Modeling in Circuit Simulation," *Computational and Applied Mathematics*, vol. 123, no. 12, pp. 395–421, 2000.

[11] MATLAB, "Documentation center," 2014.

[12] Y. M. Lee and C. C. P. Chen, "Hierarchical Model-Order Reduction for Signal-Integrity Interconnect Synthesis," in *ACM Great Lakes Symposium on VLSI*, pp. 109–114, 2001.

[13] Y. Cao, Y. M. Lee, T. H. Chen, and C. C. P. Chen, "HIPRIME: Hierarchical and Passivity Reserved Interconnect Macromodeling Engine for RLKC Power Delivery," in *ACM/IEEE Design Automation Conference*, pp. 379–384, 2002.

[14] L. Sirovich, "Turbulence and the Dynamics of Coherent Structures," *Quarterly of Applied Mathematics*, vol. 45, pp. 561–571, 1987.

[15] J. R. Phillips, "Automated Extraction of Nonlinear Circuit Macromodels," in *IEEE Custom Integrated Circuits Conference*, pp. 451–454, 2000.

[16] J. R. Phillips, "Projection Frameworks for Model Reduction of Weakly Nonlinear Systems," in *ACM/IEEE Design Automation Conference*, pp. 184–189, 2000.

[17] J. R. Wilson, *Nonlinear System Theory, the Volterra/Wiener Approach*. Johns Hopkins University Press, 1981.

[18] L. Feng, "Review of Model Order Reduction Methods for Numerical Simulation of Nonlinear Circuits," *Applied Mathematics and Computation*, vol. 167, no. 1, pp. 576 – 591, 2005.

[19] Y. Chen, "Model Order Reduction for Nonlinear Systems," m.a.sc. thesis, Massachusetts Institute For Technology, USA, 1999.

[20] C. Gu, "QLMOR: A Projection-Based Nonlinear Model Order Reduction Approach Using Quadratic-Linear Representation of Nonlinear Systems," *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 30, no. 9, pp. 1307–1320, 2011.

[21] S. A. Nahvi, M. Nabi, and S. Janardhanan, "Nonlinearity-Aware Sub-Model Combination in Trajectory based Methods for Nonlinear MOR," *Mathematics and Computers in Simulation*, vol. 94, no. 0, pp. 127 – 144, 2013.

[22] S. A. Nahvi, M. Nabi, and S. Janardhanan, "Adaptive Sampling of Nonlinear System Trajectory for Model Order Reduction," in *International Modeling, Identification, and Control Conference*, pp. 1249–1255, 2012.

[23] M. U. Farooq, L. Xia, F. A. B. Hussin, and A. S. Malik, "Automated Model Generation of Analog Circuits through Modified Trajectory Piecewise Linear Approach With ChebyShev Newton Interpolating Polynomials," in *International Conference on Intelligent Systems Modeling Simulation*, pp. 605–609, 2013.

[24] C. Gu and J. Roychowdhury, "Model Reduction via Projection onto Nonlinear Manifolds, With Applications to Analog Circuits and Biochemical Systems," in *IEEE/ACM International Conference on Computer Aided Design*, pp. 85–92, 2008.

[25] D. Dejonghe and G. Gielen, "Characterization of Analog Circuits Using Transfer Function Trajectories," *IEEE Transactions On Circuits and Systems I*, vol. 59, no. 8, pp. 1796–1804, 2012.

[26] P. Winkler, H. Aridhi, M. H. Zaki, and S. Tahar, "Generation of Reduced Analog Circuit Models Using Transient Simulation Traces," in *ACM Great Lakes Symposium on VLSI*, pp. 305–310, 2014.

[27] B. N. Bond, Z. Mahmood, Y. L., R. Sredojevic, A. Megretski, V. Stojanovi, Y. Avniel, and L. Daniel, "Compact Modeling of Nonlinear Analog Circuits Using System Identification via Semidefinite Programming and Incremental Stability Certification," *IEEE Transactions On Computer Aided Design Of Integrated Circuits And Systems*, vol. 29, no. 8, pp. 1149–1162, 2010.

[28] Y. Zhang, H. Liu, Q. Wang, N. Fong, and N. Wong, "Fast Nonlinear model Order Reduction via Associated Transforms of High Order Volterra Transfer Functions," in *ACM/IEEE Design Automation Conference*, pp. 289–294, 2012.

[29] O. Lahiouel, H. Aridhi, M. H. Zaki, and S. Tahar, "Tool for Modeling and Analysis of Analog Circuits," tech. rep., Concordia University, Montreal, QC, Canada, October 2011.

[30] P. R. Bevington and D. K. Robinson, *Data Reduction and Error Analysis for the Physical Sciences*. McGraw-Hill New York, 1969.

[31] O. Lahiouel, H. Aridhi, M. H. Zaki, and S. Tahar, "Enabling the DC Solutions Characterization Using a Fuzzy Approach," in *IEEE International Conference on New Circuits and Systems*, pp. 161–164, 2014.

[32] R. J. Baker, *CMOS Circuit Design, Layout, and Simulation*. Wiley-IEEE Press, 2010.

[33] I. Seghaier, H. Aridhi, M. H. Zaki, and S. Tahar, "A Qualitative Simulation Approach for Verifying PLL Locking Property," in *ACM Great Lakes Symposium on VLSI*, pp. 317–322, 2014.