Automated Macromodel Generation for Electronic Systems

Jaijeet Roychowdhury

ABSTRACT

Automated model generation methods are becoming an increasingly important component of methodologies for effective system verification, especially for mixed-signal electronic systems. In this paper, we organize and review the fundamental principles, development and applicability of the algorithmically-based model generation methods available today for linear and nonlinear systems.

I. INTRODUCTION

Electronic systems today, especially those for communications and sensing, are typically composed of a complex mix of digital and mixed-signal circuit blocks. Verifying such systems prior to fabrication is challenging due to their size and complexity. A common and useful approach towards verification in such situations, both during early system design and after detailed block design, is to replace large and/or complex blocks by small *macromodels* that replicate their input-output functionality well, and verify the macromodelled system. The macromodelled system can be simulated rapidly in order to evaluate different choices of design-space parameters. Such a macromodel-based verification process affords circuit and system designers considerable flexibility and convenience through the design process, especially if performed hierarchically using macromodels of differing sizes and fidelity.

The key issue in the above methodology is, of course, the creation of macromodels that represent the blocks of the system well. This is a challenging task for the wide variety of mixed-signal blocks in use today. The most prevalent approach towards creating macromodels is *manual abstraction*. Macromodels are usually created by the same person who designs the original block, often aided by simulations. While this is the only feasible approach today for many complex blocks, it does have a number of disadvantages compared to the *automated alternatives* that are the subject of this paper. Simulation often does not provide abstracted parameters of interest directly (such as poles, residues, modulation factors, *etc.*); obtaining them by manual postprocessing of simulation results is inconvenient, computationally expensive and error-prone. Manual structural abstraction of a block can easily miss the very nonidealities or interactions that detailed verification is meant to discover. With device dimensions shrinking below 100nm and non-idealities (such as substrate/interconnect coupling, degraded device characteristics, *etc.*) becoming increasingly critical, the fidelity of manually-generated macromodels to the real subsystems to be fabricated eventually is becoming increasingly suspect. Adequate incorporation of non-idealities into behavioral models, if at all possible by hand, is typically complex and laborious. Generally speaking, manual macromodelling is heuristic, time-consuming and highly reliant on detailed internal knowledge of the block under consideration, which is often unavailable when IP blocks that are not designed in-house are utilized. As a result, the potential time-to-market improvement via macromodel-based verification can be substantially negated by the time and resources needed to first generate the macromodels.

Jaijeet Roychowdhury is with the University of Minnesota, Minneapolis, USA.

It is in this context that there has been considerable interest in *automated techniques* for the creation of macromodels. Such techniques take a detailed description of a block – for example, a SPICE-level netlist – and generate, via an automated computational procedure, a much smaller macromodel. The macromodel, fundamentally a small system of equations, is usually translated into AHDL, Matlab or SPICE netlist form for use at the system level. Such an automated approach, *i.e.*, one that remains sustainable as devices shrink from deep submicron to nano-scale, is essential for realistic exploration of the design space in current and future mixed-signal SoCs/SiPs.

Several broad methodologies for automated macromodelling have been proposed. One is to generalize, abstract and automate the manual macromodelling process. For example, common topological elements in a circuit are recognized, approximated and conglomerated (*e.g.*, [11, 45]) to create a macromodel. Another class of approaches attempts to capture *symbolic* macromodels that capture the system's input-output relationship, *e.g.*, [24, 40–42, 44, 46]. Yet another class (*e.g.*, [2, 10, 15]) employs a *black-box* methodology. Data is collected via many simulations or measurements of the full system and a regression-based model created that can predict outputs from inputs. Various methods are available for the regression, including data mining, multi-dimensional tables, neural networks, genetic algorithms, *etc...*

In this paper, we focus on another methodology for macromodelling, often termed *algorithmic*[1]. Algorithmic macromodelling methods approach the problem as the transformation of a large set of mathematical equations to a much smaller one. The principal advantage of these methods is generality - so long as the equations of the original system are available numerically (*e.g.*, from within SPICE), knowledge of circuit structure, operating principles, *etc.*, is not critical. A single algorithmic method may therefore apply to *entire classes* of physical systems, encompassing circuits and functionalities that may be very disparate. Three such classes, namely linear time invariant (LTI), linear time varying (LTV), and nonlinear, are discussed in Sections II, III and IV of this paper. Algorithmic methods also tend to be more rigorous about important issues such as fidelity and stability, and often provide better guarantees of such characteristics than other methods.

II. MACROMODELLING LINEAR TIME INVARIANT (LTI) SYSTEMS

Often referred to as reduced-order modelling (ROM) or model-order reduction (MOR), automated model generation methods for Linear Time-Invariant (LTI) systems are the most mature amongst algorithmic macromodelling methods. Any block composed of resistors, capacitors, inductors, linear controlled sources and distributed interconnect models is LTI (often referred to simply as "linear"). The development of LTI MOR methods has been driven largely by the need to "compress" the huge interconnect networks, such as clock distribution nets, that arise in large digital circuits and systems. Replacing these networks by

[1]For a broader survey of macromodelling techniques, we refer the reader to, *e.g.*, [7].

small macromodels makes it feasible to complete accurate timing simulations of digital systems at reasonable computational expense. Although interconnect-centric applications have been the main domain for LTI reduction, it is appropriate for any system that is linear and time-invariant. For example, "linear amplifiers", *i.e.*, linearizations of mixed-signal amplifier blocks, are good candidates for LTI MOR methods.

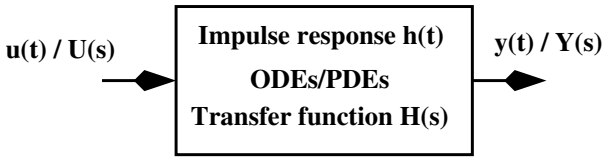

Fig. 1. Linear Time Invariant block

Figure 1 depicts the basic structure of an LTI block. $u(t)$ represents the inputs to the system, and $y(t)$ the outputs, in the time domain; in the Laplace (or frequency) domain, their transforms are $U(s)$ and $Y(s)$ respectively. The definitive property of any LTI system [48] is that the input and output are related by convolution with an *impulse response* $h(t)$ in the time domain, *i.e.*, $y(t) = x(t) * h(t)$). Equivalently, their transforms are related by multiplication with a *system transfer function* $H(s)$, *i.e.*, $Y(s) = H(s)X(s)$. Note that there may be many internal nodes or variables within the block. The goal of LTI MOR methods is to replace the block by one with far fewer internal variables, yet with an acceptably similar impulse response or transfer function.

In the majority of circuit applications, the LTI block is described to the MOR method as a set of differential equations, *i.e.*,

$$\begin{aligned} E\dot{x} &= Ax(t) + Bu(t) \\ y(t) &= C^T x(t) + Du(t) \end{aligned} \tag{1}$$

In (1), $u(t)$ represents the input waveforms to the block and $y(t)$ the outputs. Both are relatively few in number compared to the size of $x(t)$, the state of the internal variables of the block. A, B, C, D and E are constant matrices. Such differential equations can be easily formed from SPICE netlists or AHDL descriptions; especially for interconnect applications, the dimension n of $x(t)$ can be very large.

The first issue in LTI ROM is to determine what aspect of the transfer function of the original system should be retained by the reduced system; in other words, what metric of fidelity is appropriate. In their seminal 1990 paper [28], Pileggi and Rohrer used *moments* of the transfer function as fidelity metrics, to be preserved by the model reduction process. The moments m_i of an LTI transfer function $H(s)$ are related to its derivatives, *i.e.*,

$$m_1 = \left.\frac{dH(s)}{ds}\right|_{s=s_0}, \quad m_2 = \left.\frac{d^2H(s)}{ds^2}\right|_{s=s_0}, \cdots, \tag{2}$$

where s_0 is a frequency point of interest. Moments can be shown to be related to practically useful metrics, such as delay in interconnects.

In [28], Pileggi and Rohrer proposed a technique, Asymptotic Waveform Evaluation (AWE), for constructing a reduced model for the system (1). AWE first computes a number of moments of the full system (1), then uses these in another set of linear equations, the solution of which results in the reduced model. Such a procedure is termed *explicit moment matching*. The key property of AWE was that it could be shown to produce reduced models whose first several moments (at a given frequency point s_0) were identical to those of the full system. The computation involved in forming the reduced model was roughly linear in the size of the (large) original system.

While explicit moment matching via AWE proved valuable and was quickly applied to interconnect reduction, it was also observed to become numerically inaccurate as the size of the reduced model increased beyond about 10. To alleviate these, variations based on matching moments at *multiple frequency points* were proposed [1] that improved numerical accuracy. Nevertheless, the fundamental issue of numerical inaccuracy as reduced model sizes grew remained.

In 1994, Gallivan et al [5] and Feldmann/Freund [3,4] identified the reason for this numerical inaccuracy. Computing the k^{th} moment explicitly involves evaluating terms of the form $A^{-k}r$, *i.e.*, the k^{th} member of the *Krylov subspace* of A and r. If A has well separated eigenvalues (as it typically does for circuit matrices), then for $k \sim 10$ and above, only the dominant eigenvalue contributes to these terms, with non-dominant ones receding into numerical insignificance. Furthermore, even with the moments available accurately, the procedure of finding the reduced model is also poorly conditioned.

Recognizing that these are not limitations fundamental to the goal of model reduction, [3, 5] proposed alternatives. They showed that numerically robust procedures for computing Krylov subspaces, such as the Lanczos and Arnoldi (*e.g.*, [38]) methods, could be used to produce reduced models that match any given number of moments of the full system. These approaches, called *Krylov-subspace MOR techniques*, do not compute the moments of the full system explicitly at any point, *i.e.*, they perform *implicit moment matching*. In addition to matching moments in the spirit of AWE, Krylov-subspace methods were also shown to capture well the dominant poles and residues of the system. The Padé-via-Lanczos (PVL) technique [3] gained rapid acceptance within the MOR community by demonstrating its numerical robustness in reducing the DEC Alpha chip's clock distribution network.

Krylov-subspace methods are best viewed as reducing the system (1) via *projection* [6]. They produce two projection matrices, $V \in \mathcal{R}^{n\times q}$ and $W^T \in \mathcal{R}^{q\times n}$, such that the reduced system is obtained as

$$\begin{aligned} \underbrace{W^T E}_{\hat{E}}\dot{x} &= \underbrace{W^T A V}_{\hat{A}}x(t) + \underbrace{W^T B}_{\hat{B}}u(t) \\ y(t) &= \underbrace{C^T V}_{\hat{C}^T}x(t) + Du(t). \end{aligned} \tag{3}$$

For the reduction to be practically meaningful, q, the size of the reduced system, must be much smaller than n, the size of the original. If the Lanczos process is used, then $W^T V \approx I$ (*i.e.*, the two projection bases are bi-orthogonal). If the Arnoldi process is applied, then $W = V$ and $W^T V = I$.

The development of Krylov-subspace projection methods marked an important milestone in LTI macromodelling. However, reduced models produced by both AWE and Krylov methods retained the possibility of *violating passivity*, or even being *unstable*. A system is passive if it cannot generate energy under any circumstances; it is stable if for any bounded inputs, its response remains bounded. In LTI circuit applications, passivity guarantees stability. Passivity is a natural characteristic of many LTI networks, especially interconnect networks. It is essential that reduced models of these networks also be passive, since the converse implies that under some situation of connectivity, the reduced system will become unstable and diverge unboundedly from the the response of the original system.

The issue of stability of reduced models was recognized early in [5], and the superiority of Krylov-subspace methods over AWE in this regard also noted. Silveira et al [16] proposed a co-ordinate transformed Arnoldi method that guaranteed stability, but not passivity. Kerns et al [13] proposed reduction of admittance-matrix-based systems by applying a series of non-square congruence transformations. Such transformations preserve passivity properties while also retaining important poles of the system. However, this approach does not guarantee matching of system moments. A symmetric version of PVL with improved passivity and stability properties was proposed by Freund and Feldmann in 1996 [30].

The passivity-retaining properties of congruence transformations were incorporated within Arnoldi-based reduction methods for RLC networks by Odabasioglu et al [20, 21] in 1997, resulting in an algorithm dubbed PRIMA (Passive Reduced-Order Interconnect Macromodelling Algorithm). By exploiting the structure of RLC network matrices, PRIMA was able to preserve passivity *and* match moments. Methods for Lanczos-based passivity preservation [29, 47] followed.

All the above LTI MOR methods, based on Krylov-subspace computations, are efficient (*i.e.*, approximately linear-time) for reducing large systems. The reduced models produced by Krylov-subspace reduction methods are not, however, optimal, *i.e.*, they do not necessarily minimize the error for a macromodel of given size. The theory of balanced realizations, well known in the areas of linear systems and control, provides a framework in which this optimality can be evaluated. LTI reduced-order modelling methods based on *truncated* balanced realizations (TBR) (*e.g.*, [8, 9]) have been proposed. Balanced realizations are a canonical form for linear differential equation systems that "balance" controllability and observability properties. While balanced realizations are attractive in that they produce more compact macromodels for a given accuracy, the process of generating the macromodels is computationally very expensive, *i.e.*, cubic in the size of the original system. However, recent methods [17] that combine Krylov-subspace techniques with TBR methods have been successful in approaching the improved compactness of TBR, while substantially retaining the attractive computational cost of Krylov methods.

III. Macromodelling Linear Time Varying (LTV) Systems

A. Linear Time Varying (LTV) Macromodelling

LTI macromodelling methods, while valuable tools in their domain, are inapplicable to many functional blocks in mixed-signal systems, which are usually nonlinear in nature. For example, distortion or clipping in amplifiers, switching and sampling behaviour, *etc.*, cannot be captured by LTI models. In general, generating macromodels for nonlinear systems (see Section IV) is a difficult task.

However, a class of nonlinear circuits (including RF mixing, switched-capacitor and sampling circuits) can be usefully modelled as *linear time-varying* (LTV) systems. The key difference between LTV systems and LTI ones is that if the input to an LTV system is time-shifted, it does not necessarily result in the same time shift of the output. The system remains linear, in the sense that if the input is scaled, the output scales similarly. This latter property holds, at least ideally, for the input-to-output relationship of circuits such as mixers or samplers. It is the effect of a separate local oscillator or clock signal in the circuit, *independent of the signal input*, that confers the time-varying property. This is intuitive for sampling circuits, where a time-shift of the input, relative to the clock, can be easily seen not to result in the same time-shift of the original output – simply because the clock edge samples a different time-sample of the input signal. In the frequency domain, more appropriate for mixers, it is the time-varying nature that confers the key property of frequency shifting of input signals. The time-varying nature of the system can be "strongly nonlinear", with devices switching on and off – this does not impact the linearity of the signal input-to-output path.

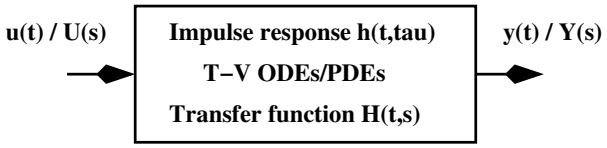

Fig. 2. Linear Time Varying block

Figure 2 depicts the basic structure of an LTV system block. Similar to LTI systems, LTV systems can also be completely characterized by impulse responses or transfer functions; however, these are now functions of two variables, the first capturing the time-variation of the system, the second the changes of the input [48]. The detailed behaviour of the system is described using time-varying differential equations, *e.g.*,

$$\begin{aligned} E(t)\dot{x} &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)^T x(t) + D(t)u(t). \end{aligned} \tag{4}$$

Time variation in the system is captured by the dependence of A, B, C, D and E on t. In many case of practical interest, this time-variation is periodic. For example, in mixers, the local oscillator input is often a sine or a square wave; switched or clocked systems are driven by periodic clocks.

The goal of macromodelling LTV systems is similar to that for LTI ones: to replace (4) by a system identical in form, but with the state vector $x(t)$ much smaller in dimension than the original. Again, the key requirement is to retain meaningful correspondence between the transfer functions of the original and reduced systems.

Because of the time-variation of the impulse response and transfer function, LTI MOR methods cannot directly be applied to LTV systems. However, Roychowdhury [33–35] showed that LTI model reduction techniques can be applied to LTV systems, by first reformulating (4) as an LTI system similar to (1), but with extra *artificial inputs* that capture the time-variation. The reformulation first separates the input and system time variations explicitly using multiple time scales [36] in order to obtain an operator expression for $H(t,s)$. This expression is then evaluated using periodic steady-state methods [14, 32, 43] to obtain an LTI system with extra artificial inputs. Once this LTI system is reduced to a smaller one using any LTI MOR technique, the reduced LTI system is reformulated back into the LTV system form (4). The use of different LTI MOR methods within this framework has been demonstrated, including explicit moment matching [33] and Krylov-subspace methods [25, 34, 35]. Moreover, Phillips [25] showed that the LTV-to-LTI reformulation could be performed using standard linear system theory concepts [48], without the use of multiple time scales.

IV. Macromodelling Nonlinear Systems

While wires, interconnect, and passive lumped elements are purely linear, any mixed-signal circuit block containing semiconductor devices is nonlinear. Nonlinearity is, in fact, a fundamental feature of any block that provides signal gain, or performs any function more complex than linear filtering. Even

though linear approximations of many nonlinear blocks are central to their design and intended operation, it is usually important to consider the impact of nonlinearities with a view to limiting their impact. For example, in "linear" amplifiers and mixers, distortion and intermodulation, caused solely by nonlinearities, must typically be guaranteed not to exceed a very small fraction of the output of the linearized system. This is especially true for traditional RF and microwave designs. Such *weakly nonlinear systems* comprise an important class of blocks that can benefit from macromodelling.

Additionally, many nonlinear blocks of interest are not designed to be approximately linear in operation. Examples include digital gates, switches, comparators, *etc.*, which are intended to switch abruptly between two states. While such operation is obviously natural for purely digital systems, strongly nonlinear behaviour is also exploited in analog blocks such as sampling circuits, switching mixers, analog-to-digital converters *etc.*. Furthermore, oscillators and PLLs, which are common and basic components in mixed-signal systems, exhibit complex dynamics which are fundamentally strongly nonlinear.

Unlike for the classes of linear systems considered in the previous sections, no technique currently exists that is capable, even in principle, of producing a macromodel that conforms to any reasonable fidelity metric for *completely general* nonlinear systems. The difficulty stems from the fact that nonlinear systems are richly varied, with extremely complex dynamical behaviour possible that is very far from being exhaustively investigated or understood. This is in contrast to linear dynamical systems, for which comprehensive mathematical theories exist (see, *e.g.*, [48]) that are universally applicable. In view of the diversity and complexity of nonlinear systems in general, it is difficult to conceive of a single overarching theory or method that can be employed for effective macromodelling of an arbitrary nonlinear block. It is not surprising, therefore, that macromodelling of nonlinear systems has tended to be manual, relying heavily on domain-specific knowledge for specialized circuit classes, such as ADCs, phase detectors, *etc.*.

In recent years, however, linear macromodelling methods have been extended to handle weakly nonlinear systems. Other techniques based on piecewise approximations have also been devised that are applicable some strongly nonlinear systems. As described below in more detail, these approaches start from a general nonlinear differential equation description of the full system, but first approximate it to a more restrictive form, which is then reduced to yield a macromodel of the same form. The starting point is a set of nonlinear differential-algebraic equations (DAEs) of the form

$$\begin{aligned} \dot{q}(x(t)) &= f(x(t)) + b\,u(t) \\ y(t) &= c^T x(t), \end{aligned} \tag{5}$$

where $f(\cdot)$ and $q(\cdot)$ are nonlinear vector functions.

A. Polynomial-based weakly nonlinear methods

To appreciate the basic principles behind weakly nonlinear macromodelling, it is first necessary to understand how the full system can be treated if the nonlinearities in (5) are approximated by low-order polynomials. The polynomial approximation concept is simply an extension of linearization, with $f(x)$ and $q(x)$ replaced by the first few terms of a Taylor series about an expansion point x_0 (typically the DC solution); for example,

$$f(x) = f(x_0) + A_1(x-x_0) + A_2(x-x_0)^{\textcircled{2}} + \cdots, \tag{6}$$

where $a^{\textcircled{i}}$ represents the Kronecker product of a with itself i times. When (6) and its $q(\cdot)$ counterpart are used in (5), a system of polynomial differential equations results. If $q(x) = x$ (assumed for simplicity), these equations are of the form

$$\begin{aligned} \dot{x}(t) &= f(x_0) + A_1(x-x_0) + A_2(x-x_0)^{\textcircled{2}} + \cdots + b\,u(t) \\ y(t) &= c^T x(t). \end{aligned} \tag{7}$$

The utility of this polynomial system is that it becomes possible to leverage an existing body of knowledge on weakly polynomial differential equation systems, *i.e.*, systems where the higher-order nonlinear terms in (6) are small compared to the linear term. In particular, *Volterra series theory* [39] and weakly-nonlinear perturbation techniques [19] justify a relaxation-like approach for such systems, which proceeds as follows. First, the response of the linear system, ignoring higher-order polynomial terms, is computed – denote this response by $x_1(t)$. Next, $x_1(t)$ is inserted into the quadratic term $A_2(x-x_0)^{\textcircled{2}}$ (denoted a *distortion input*), the original input is *substituted* by this waveform, and the *linear* system solved again to obtain a *perturbation due to the quadratic term* – denote this by $x_2(t)$. The sum of x_1 and x_2 is then substituted into the cubic term to obtain another weak perturbation, the linear system solved again for $x_3(t)$, and so on. The final solution is the sum of x_1, x_2, x_3 and so on. An attractive feature of this approach is that the perturbations x_2, x_3, *etc.*, which are available *separately* in this approach, correspond to quantities like distortion and intermodulation which are of interest in design. Note that at every stage, to compute the perturbation response, a *linear* system is solved – nonlinearities are captured via the distortion inputs to these systems.

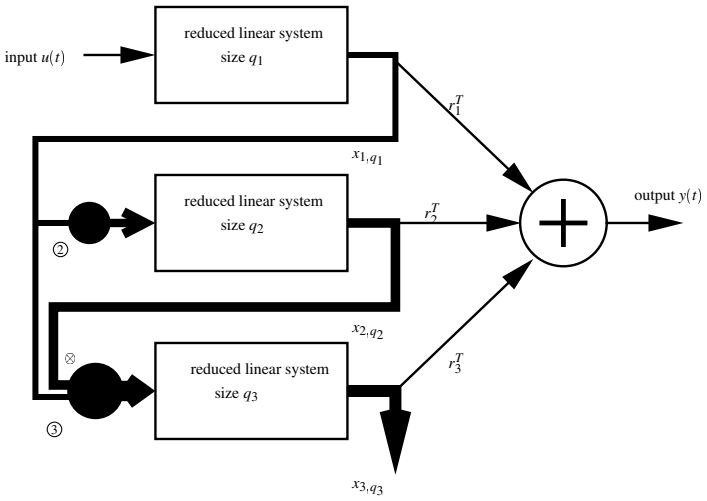

Fig. 3. Block structure of reduced polynomial system

The basic idea behind macromodelling weakly nonlinear systems is to exploit this fact; in other words, to apply linear macromodelling techniques, appropriately modified to account for distortion inputs, to each stage of the relaxation process above. In the first such approach, proposed in 1999 by Roychowdhury [35], the linear system is first reduced by LTI MOR methods to a system of size q_1, as shown in Figure 3, via a projection basis obtained using Krylov-subspace methods. The distortion inputs for the quadratic perturbation system are then expressed in terms of the *reduced* state vector of the linear term, to obtain an input vector of size q_1^2 . The quadratic perturbation system (which has the same linear system matrix, but a different input vector) is then again reduced via another projection basis,

to size q_2. This process is continued for higher order terms. The overall reduced model is the union of the separate reduced models with outputs summed together, as depicted in Figure 3.

By tailoring projection bases for each nonlinearly-perturbed linear system, this approach focusses on accuracy; however, this is achieved at the cost of increased macromodel size $q_1 + q_2 + \cdots$. Recognizing the size issue, Phillips in 2000 [26, 27] proposed that a *single* projection basis be applied to the system (7) (analogous to LTI MOR systems), and also observed that Carlemann bilinearization [37] could be employed to obtain a canonical equation form. Intuitively, the use of a single projection basis consolidates the commonality in the three reduced models shown in Figure 3, leading to smaller overall models.

In 2003, Li and Pileggi proposed the NORM method [23], which combines and extends the above two approaches. Similar to [35], NORM generates tailored projection bases for each perturbed linear system, but instead of retaining separate macromodels as in Figure 3, it compresses these projection bases into a single projection basis. NORM then employs this single projection basis to reduce the system (7) as proposed in [27]. A particularly attractive property of NORM is that it produces a macromodel that matches a number of *multidimensional moments* of the Volterra series kernels [39] of the system – indeed, the distortion terms for each perturbed system are pruned to ensure matching of a specified number of moments. The authors of NORM also include a variant that matches moments at multiple frequency points.

B. Piecewise approximation methods

The polynomial approximations discussed above are excellent when the intended operation of the system exercises only weak nonlinearities, as in power amplifiers, "linear" mixers, *etc.*. Outside a relatively small range of validity, however, polynomials are well known to be extremely poor global approximators. This limitation is illustrated in Figure 4, where it can be seen that, outside a local region where there is a good match, even a sixth-degree Taylor-series approximation diverges dramatically from the function it is meant to represent.

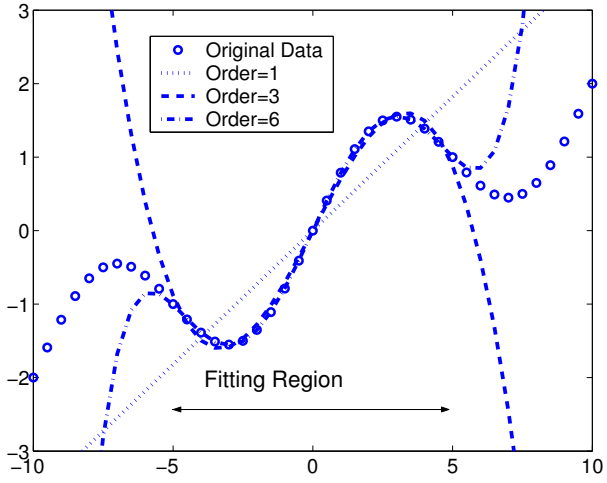

Fig. 4. Limitations of global polynomial approximations

It is for this reason that other ways of approximating (5) that have better global approximation properties than polynomials have been sought. One approach is to represent the nonlinear functions $f(\cdot)$ and $q(\cdot)$ in (5) by *piecewise linear (PWL)* segments. The state space is split into a number of disjoint regions, and within each region, a linear approximation is used that matches the nonlinear function approximately within the region. By using a sufficiently large number of regions, the nonlinear function can be represented accurately over the entire domain of interest. From a macromodelling perspective, the motivation for PWL approximations is that since the system is linear within each region, linear macromodelling methods can be leveraged.

Piecewise linear approximations are not new in circuit simulation, having been employed in the past most notably in attempts to solve the DC operating point problem [12, 22]. One concern with these methods is a potential exponential explosion in the number of regions as the dimension of the state space grows. This is especially the case when each elemental device within the circuit is first represented in piecewise form, and the system of circuit equations constructed from these piecewise elements. A combinatorial growth of polytope regions results, via cross-products of the hyperplanes that demarcate piecewise regions within individual devices.

To circumvent the explosion of regions, which would severely limit the simplicity of a small macromodel, Rewienski and White proposed the Trajectory PWL method (TPWL) [31] in 2001. In TPWL, a reasonable number of "center points" is first selected along a simulation trajectory in the the state space, generated by exciting the circuit with a representative training input. Around each center point, system nonlinearities are approximated by linearization, with the region of validity of the linearization defined *implicitly*, as consisting of all points that are closer to the given center point than to any other. Thus there are only as many piecewise regions as center points, and combinatorial explosion resulting from intersections of hyperplanes is avoided. The implicit piecewise regions in TPWL are in fact identical to the Voronoi regions defined by the collection of center points chosen.

Within each piecewise region, the TPWL approach simply reduces the linear system using existing LTI MOR methods to obtain a reduced linear model. The reduced linear models of all the piecewise regions are finally stitched together using a *scalar weight function* to form a single-piece reduced model. The weight function identifies, using a closest-distance metric, whether a test point in the state space is within a particular piecewise region, and weights the corresponding reduced linear system appropriately.

The TPWL method, by virtue of its use of inherently better PWL global approximation, avoids the blow-up that occurs when polynomial-based methods are used with large inputs. It is thus better suited for circuits with strong nonlinearities, such as comparators, digital gates, *etc.*.. However, because PWL approximations do not capture higher-order derivative information, TPWL's ability to reproduce small-signal distortion or intermodulation is limited.

To address this limitation, Dong and Roychowdhury proposed a piecewise polynomial (PWP) extension [18] of TPWL in 2003. PWP combines weakly nonlinear MOR techniques with the piecewise idea, by approximating the nonlinear function in each piecewise region by a polynomial, rather than a purely linear, Taylor expansion. Each piecewise polynomial region is reduced using one of the polynomial MOR methods outlined above, and the resulting polynomial reduced stitched together with a scalar weight function, similar to TPWL. Thanks to its piecewise nature, PWP is able to handle strong nonlinearities globally; because of its use of local Taylor expansions in each region, it is also able to capture small-signal distortion and intermodulation well. Thus PWP expands the scope of applicability of nonlinear macromodelling to encompass blocks in which strong and weak nonlinearities both play an important

functional rôle.

V. Conclusion

Automated bottom-up macromodelling is rapidly becoming critical for the effective hierarchical verification of large mixed-signal systems. We have discussed the main algorithmic macromodelling approaches available today. Linear time-invariant methods, the subject of research for more than a decade, have already proven their usefulness for interconnect analysis. Issues such as the fidelity, compactness, dynamical stability and passivity of generated macromodels have been identified and addressed. Extensions to linear time-varying systems, useful for mixers and sampling circuits, have also been demonstrated to produce useful, compact models. Interest in macromodelling nonlinear systems has grown rapidly over the last few years and a number of promising approaches have emerged. It is likely that further research in automated nonlinear macromodelling will translate into practically useful tools in the near future.

Acknowledgments

We would like to thank Joel Phillips, Jacob White, Larry Pileggi, and Ken Kundert for useful discussions related to macromodelling. Special thanks are due to Ning Dong for many valuable discussions and for permission to use Figure 4.

References

[1] E. Chiprout and M.S. Nakhla. *Asymptotic Waveform Evaluation*. Kluwer, Norwell, MA, 1994.

[2] D. Schreurs, J. Wood, N. Tufillaro, D. Usikov, L. Barford, and D.E. Root. The construction and evaluation of behavioral models for microwave devices based on time-domain large-signal measurements. In *Proc. IEEE IEDM*, pages 819–822, December 2000.

[3] P. Feldmann and R.W. Freund. Efficient linear circuit analysis by Padé approximation via the Lanczos process. *IEEE Trans. CAD*, 14(5):639–649, May 1995.

[4] R.W. Freund and P. Feldmann. Efficient Small-signal Circuit Analysis And Sensitivity Computations With The Pvl Algorithm. *Proc. ICCAD*, pages 404–411, November 1995.

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

[6] E.J. Grimme. *Krylov Projection Methods for Model Reduction*. PhD thesis, University of Illinois, EE Dept, Urbana-Champaign, 1997.

[7] H.A. Mantooth, L. Ren, X. Huang, Y. Feng, and W. Zheng. A survey of bottom-up behavioral modeling methods for analog circuits. In *Proc ISCAS*, pages 910–913, May 2003.

[8] J. Phillips, L. Daniel, and L.M. Silveira. Guaranteed passive balancing transformations for model order reduction. In *Proc. IEEE DAC*, pages 52–57, June 2002.

[9] J-R. Li and J. White. Efficient model reduction of interconnect via approximate system gramians. In *Proc. ICCAD*, pages 380–383, November 1999.

[10] J. Wood and D.E. Root. The behavioral modeling of microwave/RF ICs using non-linear time series analysis. In *IEEE MTT-S Digest*, pages 791–794, June 2003.

[11] K. Francken, M. Vogels, E. Martens, and G. Gielen. A behavioral simulation tool for continuous-time /spl Delta//spl Sigma/ modulators. In *Proc. ICCAD*, pages 229–233, November 2002.

[12] J. Katzenelson and L.H. Seitelman. An iterative method for solution of nonlinear resistive networks. Technical Report TM65-1375-3, AT&T Bell Laboratories.

[13] K.J. Kerns, I.L. Wemple, and A.T. Yang. Stable and efficient reduction of substrate model networks using congruence transforms. In *Proc. ICCAD*, pages 207–214, November 1995.

[14] K.S. Kundert, J.K. White, and A. Sangiovanni-Vincentelli. *Steady-state methods for simulating analog and microwave circuits*. Kluwer Academic Publishers, 1990.

[15] L. Hongzhou, A. Singhee, R. Rutenbar, and L.R. Carley. Remembrance of circuits past: macromodeling by data mining in large analog design spaces. In *Proc. IEEE DAC*, pages 437–442, June 2002.

[16] L.M. Silveira, M. Kamon, I. Elfadel, and J. White. A coordinate-transformed Arnoldi algorithm for generating guaranteed stable reduced-order models of RLC circuits. In *Proc. ICCAD*, pages 288–294, November 1996.

[17] M. Kamon, F. Wang, and J. White. Generating nearly optimally compact models from Krylov-subspace based reduced-order models. *IEEE Trans. Ckts. Syst. – II: Sig. Proc.*, pages 239–248, April 2000.

[18] N. Dong and J. Roychowdhury. Piecewise Polynomial Model Order Reduction. In *Proc. IEEE DAC*, June 2003.

[19] A. Nayfeh and B. Balachandran. *Applied Nonlinear Dynamics*. Wiley, 1995.

[20] A. Odabasioglu, M. Celik, and L.T. Pileggi. PRIMA: passive reduced-order interconnect macromodelling algorithm. In *Proc. ICCAD*, pages 58–65, November 1997.

[21] A. Odabasioglu, M. Celik, and L.T. Pileggi. PRIMA: passive reduced-order interconnect macromodelling algorithm. *IEEE Trans. CAD*, pages 645–654, August 1998.

[22] Fujisawa Ohtsuki and Kumagai. Existence theorems and a solution algorithm for piecewise linear resistor networks. *SIAM J. Math. Anal.*, 8, February 1977.

[23] P. Li and L. Pileggi. NORM: Compact Model Order Reduction of Weakly Nonlinear Systems. In *Proc. IEEE DAC*, pages 472–477, June 2003.

[24] P. Vanassche, G. Gielen, and W. Sansen. Constructing symbolic models for the input/output behavior of periodically time-varying systems using harmonic transfer matrices . In *Proc. IEEE DATE Conference*, pages 279–284, March 2002.

[25] J. Phillips. Model Reduction of Time-Varying Linear Systems Using Approximate Multipoint Krylov-Subspace Projectors. In *Proc. ICCAD*, November 1998.

[26] J. Phillips. Projection-based approaches for model reduction of weakly nonlinear, time-varying systems. *IEEE Trans. CAD*, 22(2):171–187, February 2000.

[27] J. Phillips. Projection frameworks for model reduction of weakly nonlinear systems. In *Proc. IEEE DAC*, June 2000.

[28] L.T. Pillage and R.A. Rohrer. Asymptotic waveform evaluation for timing analysis. *IEEE Trans. CAD*, 9:352–366, April 1990.

[29] R. Freund. Passive reduced-order models for interconnect simulation and their computation via Krylov-subspace algorithms. In *Proc. IEEE DAC*, pages 195–200, June 1999.

[30] R. Freund and P. Feldmann. Reduced-order modeling of large passive linear circuits by means of the SyPVL algorithm. In *Proc. ICCAD*, pages 280–287, November 1996.

[31] M. Rewienski and J. White. A Trajectory Piecewise-Linear Approach to Model Order Reduction and Fast Simulation of Nonlinear Circuits and Micromachined Devices. In *Proc. ICCAD*, November 2001.

[32] M. Rösch and K.J. Antreich. Schnell stationäre simulation nichtlinearer schaltungen im frequenzbereich. *AEÜ*, 46(3):168–176, 1992.

[33] J. Roychowdhury. MPDE methods for efficient analysis of wireless systems. In *Proc. IEEE CICC*, May 1998.

[34] J. Roychowdhury. Reduced-order modelling of linear time-varying systems. In *Proc. ICCAD*, November 1998.

[35] J. Roychowdhury. Reduced-order modelling of time-varying systems. *IEEE Trans. Ckts. Syst. – II: Sig. Proc.*, 46(10), November 1999.

[36] J. Roychowdhury. Analysing circuits with widely-separated time scales using numerical PDE methods. *IEEE Trans. Ckts. Syst. – I: Fund. Th. Appl.*, May 2001.

[37] W. Rugh. *Nonlinear System Theory - The Volterra-Wiener Approach*. Johns Hopkins Univ Press, 1981.

[38] Y. Saad. *Iterative methods for sparse linear systems*. PWS, Boston, 1996.

[39] M. Schetzen. *The Volterra and Wiener Theories of Nonlinear Systems*. John Wiley, 1980.

[40] S.X.-D Tan and C.J.-R Shi. Efficient DDD-based term generation algorithm for analog circuit behavioral modeling. In *Proc. IEEE ASP-DAC*, pages 789–794, January 2003.

[41] S.X.-D Tan and C.J.-R Shi. Efficient DDD-based term generation algorithm for analog circuit behavioral modeling. In *Proc. IEEE DATE Conference*, pages 1108–1009, March 2003.

[42] S.X.-D Tan and C.J.-R Shi. Efficient very large scale integration power/ground network sizing based on equivalent circuit modeling. *IEEE Trans. CAD*, 22(3):277–284, March 2003.

[43] R. Telichevesky, K. Kundert, and J. White. Efficient steady-state analysis based on matrix-free krylov subspace methods. In *Proc. IEEE DAC*, pages 480–484, 1995.

[44] W. Daems, G. Gielen, and W. Sansen. A fitting approach to generate symbolic expressions for linear and nonlinear analog circuit performance characteristics. In *Proc. IEEE DATE Conference*, pages 268–273, March 2002.

[45] X. Huang, C.S. Gathercole, and H.A. Mantooth. Modeling nonlinear dynamics in analog circuits via root localization. *IEEE Trans. CAD*, 22(7):895–907, July 2003.

[46] Y. Qicheng and C. Sechen. A unified approach to the approximate symbolic analysis of large analog integrated circuits. *IEEE Trans. Ckts. Syst. – I: Fund. Th. Appl.*, pages 656–669, August 1996.

[47] Z. Bai, R. Freund, and P. Feldmann. How to make theoretically passive reduced-order models passive in practice. In *Proc. IEEE CICC*, pages 207–210, May 1998.

[48] L. A. Zadeh and C. A. Desoer. *Linear System Theory: The State-Space Approach*. McGraw-Hill Series in System Science. McGraw-Hill, New York, 1963.