2005 International Symposium on Nonlinear
Theory and its Applications (NOLTA2005)
Bruges, Belgium, October 18-21, 2005

Algorithms for the Efficient Computation and Application of Volterra Kernels in the Behavior Analysis of Periodically Excited Nonlinear Dynamical Circuits and Systems

Andreas Bauer† and Wolfgang Schwarz‡

†**ZMD AG**, BU ASSP, BL Opto/IR
Grenzstr. 28, 01109 Dresden, Germany
email: bauer@zmd.de

‡TU Dresden, Faculty of Electrical Engineering/IEE
Helmholtzstrae 10, 01062 Dresden, Germany
email: schwarz@iee1.et.tu-dresden.de

Abstract— This paper provides an overview about a class of recently developed algorithms for the computation, handling and application of Volterra Series in the analysis of nonlinear circuits and systems. The idea of the presented methods is based on the representation of products of exponential functions by multi-indexes. It is illustrated how these multi-indexes enable efficient operations without tedious formula manipulations for the computation and application of Volterra Kernels in the behavior description and analysis of nonlinear dynamical circuits and systems. An implemtation of a program toolbox framework in Mathematica and C is presented in brief.

1. Introduction

The behavior analysis of nonlinear dynamical circuits and systems is an important tool for the development and parametrical optimization of information processing circuits or systems. The success of these analysis depends on the availability of a suitable behavior description. A behavior description represents the behavior of the system in a mathematical way. It is parameterized by the system parameters. Applying mathematical methods and techniques to these behavior descriptions allows the parametrical analysis or optimization of the system with regard to specified behavior characteristics.

Thus, the key for the behavior analysis of the system is its behavior description. Provided a sufficient accuracy, behavior descriptions which approximately represent the system behavior can be used in order to simplify and accelerate the analysis.

Many analysis of information processing circuits and systems consider the input-output behavior of the system. A suitable behavior description should allow the efficient computation of steady state responses of the system to periodic input signals.

In several publications e.g. [CN79a, CN79b, WS98] the application of Volterra series in the transform domain as a behavior description is suggested. Volterra series allow an efficient computation of an approximative steady state response of weakly nonlinear circuits and systems.

Here, an approach to the computation and handling of Volterra kernels and series based on multi-indexes is presented. This approach tries to overcome the tedious formula manipulations involved in the computation and application of Volterra kernels and series. It is a possible key for algorithms for the automatic computation and handling of Volterra kernels.

The paper will concentrate on the description of the multi-index algorithm development for the growing exponential method (GEM) for nonlinear systems. An example is used to ilustrate the approach. The extension of the idea of the algorithm to nonlinear networks is presented only at a glance.

2. Explicite behavior description using Volterra Kernels

When a behavior description of a given structure such as a circuit netlist or a system diagram is derived in most cases an systeme of nonlinear ordinary differential equations is obtained. ODEs are an *implicit* form of an behavior description. Fig. 1 shows a system diagram of such an implicit behavior description. This

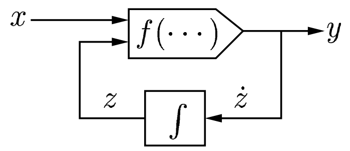

Figure 1: Implicit behavior description diagram

behavior description is given by the equations

$$\begin{aligned} \dot{z}(t) &= f(z(t), x(t)) \\ y(t) &= g(z(t), x(t)) \end{aligned} \qquad (1)$$

Here $g()$ is simply $g(z(t), x(t)) = I(x(t))$ with $I()$ being the identity. If the system of ODEs in (1) is nonlinear, for almost all systems and input signals x the map $\varphi : x \in \mathcal{X} \mapsto y \in \mathcal{Y}$ can not be found in closed analytical form. Here $\mathcal{X}$ is the set of input signals. Respectively, $\mathcal{Y}$ is the set of output signals with an element y.

Thus, in the practical analysis of nonlinear circuits and systems the mapping φ is computed numerically for a limited time interval T and for a particular setting of system parameters, initial conditions and one input signal. Such an analysis is time and memory consuming and the separation of the transient and steady state behavior is a complicated task.

For most analysis problems it is advantageous to use an explicit behavior description. A diagram of such an explicit behavior description is depicted in Fig. 2.

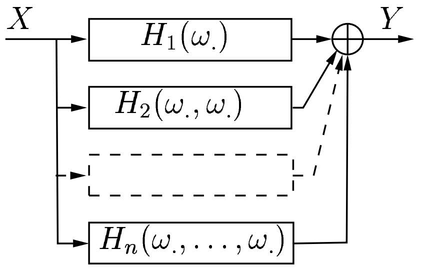

Figure 2: A Volterra system in the frequency domain as an explicit behavior description

In the diagram the behavior description is given in the frequency domain. Thus, the class of input signals is limited to periodic signals. Instead of sampled waveforms $y_i = y(t_i), i = 0, 1, \dots$ as in the case of implicit behavior descriptions, complex amplitudes $\underline{\mathrm{Y}} = \hat{\mathrm{Y}} \mathrm{e}^{\mathrm{j}\,\varphi_Y}$ are computed here. The computation involves straight forward algebraic operations only and thus is very efficient. Such a behavior description is very well suited for an wide class of analysis, called nonlinear small signal analysis.

A possible explicit behavior description as shown is Fig. 2 is obtained when using *Volterra series* in the frequency domain. The blocks in Fig. 2 are the Volterra kernels $\underline{\mathrm{H}}_n(\omega_1, \dots, \omega_n)$ of order n. These Kernels are parameterized by the system parameters $\boldsymbol{P}$ and the frequencies ω_i of the input signal.

In order to obtain a behavior description of the form depicted in Fig. 2, the Volterra kernels $\underline{\mathrm{H}}_n(\omega_1, \dots, \omega_n)$ have to be computed from an implicit behavior description of the form (1). For this task several methods have been developed. Such methods are described for system equations in [Rug81] and for networks in [WS98] for example.

Common to these methods is the first analysis step. The general implicit behavior description

$$\boldsymbol{f}(x, \dot{x}, \ddot{x}, \dots, y, \dot{y}, \ddot{y}, \dots) = 0 \qquad (2)$$

has to be transformed into an approximative polynomial form with a finite order about an operation point. If the operating point is assumed to be known as $x = x_0, y = y_0$ the approximative implicit behavior description is

$$\mathcal{P}_f|_{x=x_0, y=y_0} = P_f(x, \dot{x}, \ddot{x}, \dots, y, \dot{y}, \ddot{y}, \dots) = 0\,. \qquad (3)$$

It is not a subject of this paper how the polynomial form (3) is computed from (2). However, the same computation step is involved in the *linear* small signal analysis. There (3) is truncated after the first order terms of the polynomial series. Thus at least the operating point computation and the computation of the first order terms in the Volterra kernel algorithm is available.

The multi-index algorithm has been developed for two Volterra kernel computation methods, for the *Growing Exponential Method* (GEM) [Rug81] and for the *Volterra Nodal Voltage Analysis* (VNVA) [WS98].

The basic idea of the multi-index algorithms arises form a closer look at the GEM. The method is based on ansatzes x_A and y_A for the input and output signals.

$$\begin{aligned} x_A(t) &= \mathrm{e}^{\lambda_1 t} + \mathrm{e}^{\lambda_2 t} + \dots + \mathrm{e}^{\lambda_N t} = \sum_{\boldsymbol{m}^{N,1}} \mathrm{e}^{\boldsymbol{m}_i^{N,1} \Lambda t} \\ y_A(t) &= \mathrm{G}_{\boldsymbol{m}_1^N} \mathrm{e}^{\boldsymbol{m}_1^N \Lambda t} + \dots \mathrm{G}_{\boldsymbol{m}_I^N} \mathrm{e}^{(\boldsymbol{m}_I^N \Lambda) t} \end{aligned} \qquad (4)$$

with $\Lambda = (\lambda_1, \dots, \lambda_N)$. In (4) $\boldsymbol{m}_i^N$ are multi-indexes of the multi-index set $\boldsymbol{m}^N$. The elements of this set are

$$\boldsymbol{m}^{N,k} = (m_1^{N,k}, \dots, m_n^{N,k}) = (\overbrace{\underbrace{1, \dots, 1}_{k}, 0, \dots, 0}^{N}) \qquad (5)$$

with $k = 1, 2, \dots, N$ and all their possible permutations. For example, the $\boldsymbol{m}^3$ is the set

$$\begin{aligned} \boldsymbol{m}^3 = \{&(1,0,0), (0,1,0), (0,0,1), (1,1,0), \\ &(1,0,1), (0,1,1), (1,1,1)\} \end{aligned} \qquad (6)$$

The ansatzes (4) replace $x \to x_A$ and $y \to y_A$ in (3). The equation is expanded and solved for the $\mathrm{G}_{\boldsymbol{m}^{N,k}}$ of (4) according to (5) by means of a comparison of coefficients of the exponential functions $\mathrm{e}^{\boldsymbol{m}^{N,k} \Lambda t}$. Thus, in order to find $\mathrm{G}_{m^{N,k}}$ all summands containing the exponential $\mathrm{e}^{m^{N,k} \Lambda t}$ are collected. The resulting equations are solved for the $\mathrm{G}_{m^{N,k}}$ which are proportional to the Volterra kernels $\mathrm{H}_k(\omega_1, \dots, \omega_k)$ where the replacements $\lambda_i \to \omega_i$ were applied.

The straight forward use of this method results in tedious formula manipulations with a huge amount of terms. Since many of these generated terms do not contribute to $\mathrm{G}_{m^{N,k}}$, the computational effort is unnecessarily high.

When the ansatzes are replaced into (3) and the equation is expanded terms of the form

$$\prod_L x \cdot \prod_M y \;\rightarrow\; P \cdots \mathrm{e}^{m_l^{N,1}\Lambda t} \cdot \mathrm{e}^{m_m^N \Lambda t} \cdots \tag{7}$$

are obtained, where P is the parameter to the corresponding product. Equation (7) can be rewritten as:

$$\begin{aligned}\prod_L x \cdot \prod_M y \;\rightarrow\;\; & P\,\mathrm{e}^{(\cdots + m_l^{N,1}\Lambda t + m_m^N \Lambda t + \cdots)} = \\ & P\,\mathrm{e}^{(\cdots + m_l^{N,1} + m_m^N + \cdots)\Lambda t}\end{aligned} \tag{8}$$

As stated above, the comparison of coefficients is performed for coefficients at the exponential $\mathrm{e}^{\boldsymbol{m}^{N,k}\Lambda t}$. Thus, for the selection of the needed terms it is sufficient to check if the condition

$$\boldsymbol{m}^{N,k} \overset{?}{=} \cdots + m_l^{N,1} + m_m^N + \cdots \tag{9}$$

is true for the considered sum of multi-indexes $\cdots + m_l^{N,1} + m_m^N + \cdots$. If (9) is true the term contributes to $\mathrm{G}_{\boldsymbol{m}^{N,k}}$ and therefore to $\mathrm{H}_k(\ldots)$. These multi-indexes are used for the construction of terms for the determining equation of $\mathrm{G}_{\boldsymbol{m}^{N,k}}$. All other terms do not need to be constructed.

The above approach allows use the multi-index sets $y \rightarrow \boldsymbol{m}^N$ and $x \rightarrow \boldsymbol{m}^{N,1}$ instead of the ansatzes (4) in the calculation. Only in the last stage of the calculation, the equation to be solved is constructed from the remaining $L + M$-tuples of multi-indexes.

In order to simplify the calculations further, in the beginning of the algorithm structure and parameters P of (3) are separated according to:

$$\begin{aligned}&\mathcal{P}_f(x, \dot{x}, \ddot{x}, \ldots, y, \dot{y}, \ddot{y}, \ldots, \boldsymbol{p}) \\ &\quad = \Pi(x, \dot{x}, \ddot{x}, \ldots, y, \dot{y}, \ddot{y}, \ldots) \cdot \mathrm{P}(\mathrm{K}, \mathrm{p})\end{aligned} \tag{10}$$

where $\Pi(\ldots)$ is a vector of all products of the input and output signals and their time derivatives. $\mathrm{P}(\mathrm{K}, \mathrm{p})$ is the vector of constants K and system parameters p belonging to the products in $\Pi(\ldots)$.

In the next step of the algorithm the multi-index sets $\boldsymbol{m}_y = \boldsymbol{m}^N$ and $\boldsymbol{m}_x = \boldsymbol{m}^{N,1}$ will be constructed where N is the order of the Volterra kernel to be computed.

Next all products of $\Pi(\ldots)$ with more than N factors are deleted from $\Pi(\ldots)$, because (9) can not be satisfied if more than N multi-indexes are added.

The remaining products are replaced by Cartesian products of multi-index sets according to:

$$\underbrace{x \cdot \dot{x} \cdot \cdots}_{L} \cdot \underbrace{y \cdot \dot{y} \cdots}_{M} \Rightarrow \boldsymbol{m}_x \times \boldsymbol{m}_x \times \cdots \boldsymbol{m}_y \times \boldsymbol{m}_y \times \cdots \tag{11}$$

In the place of every product in $\Pi(\ldots)$ is now a set of $L + M$-tuples of multi-indexes. It should be noted that the replacement is done regardless whether the replaced factor is a time derivative or not.

In the next step all $L + M$-tuples of multi-indexes undergo the test (9). If (9) is true the $L + M$-tuple remains in the set, otherwise it is deleted from the list.

From all remaining $L + M$-tuples multi-indexes in the components of $\Pi(\ldots)$ products will be constructed. The factors are constructed according to:

$$\boldsymbol{m}_* \rightarrow \begin{cases} \boldsymbol{m}_{*,1}\lambda_1^d + \cdots + \boldsymbol{m}_{*,N}\lambda_N^d & \text{if } \boldsymbol{m}_* \Rightarrow x^{(d)} \\ (\boldsymbol{m}_{*,1}\lambda_1 + \cdots + \boldsymbol{m}_{*,N}\lambda_N)^d \mathrm{G}_{\boldsymbol{m}_*} & \text{if } \boldsymbol{m}_* \Rightarrow y^{(d)} \end{cases} \tag{12}$$

In (12) the superscript (d) denotes the order of the time derivative of the corresponding factor. The resulting products within each component of $\Pi(\ldots)$ are added and (10) is applied with the replaced $\Pi(\ldots)$ resulting in the equation to be solved for $\mathrm{G}_{\boldsymbol{m}^{N,k}}$. The solution of $\mathrm{G}_{\boldsymbol{m}^{N,k}}$ contains lower order $\mathrm{G}_{\boldsymbol{m}^{N,k'}}$ with $k' < k$. These $\mathrm{G}_{\boldsymbol{m}^{N,k'}}$ are obtained with same algorithm.

In [Bau04] the algorithm is described in detail. Additionally the algorithm is extended there in order to circumvent the symbolic solution for $\mathrm{G}_{\boldsymbol{m}^{N,k}}$. In the extended algorithm numerator and denominator of $\mathrm{G}_{\boldsymbol{m}^{N,k}}$ are constructed directly from $\Pi(\ldots)$.

2.1. Example

In order to illustrate the algorithm the Volterra kernels of order 1 and 2 are computed for a simple system with quadratic nonlinearity:

$$a_1 x - a_1 y + a_2 x^2 - 2a_2 xy + a_2 y^2 - C\dot{y} = 0 \tag{13}$$

The decomposition according to (10) yields:

$$\begin{aligned}\Pi &= (x, y, xx, xy, yy, \dot{y}) \\ \mathrm{P}(\mathrm{K}, \mathrm{p}) &= (a_1, -a_1, a_2, -2a_2, a_2, -C)\end{aligned} \tag{14}$$

The constructed sets of multi-indexes are:

$$\begin{aligned}\boldsymbol{m}_x &= \{(1,0), (0,1)\} \\ \boldsymbol{m}_y &= \{(1,0), (0,1), (1,1)\}\end{aligned} \tag{15}$$

Replacing the factors in Π of (14) with (15) according to (11) yields the following vector of multi-indexes and pairs of multi-indexes:

$$\begin{aligned}&\Pi = (\boldsymbol{m}_x, \boldsymbol{m}_y, \boldsymbol{m}_x \times \boldsymbol{m}_x, \boldsymbol{m}_x \times \boldsymbol{m}_y, \boldsymbol{m}_y \times \boldsymbol{m}_y, \boldsymbol{m}_y\ ) \\ &= (\{\{(1,0)\}, \{(0,1)\}\}, \{\{(1,0)\}, \{(0,1)\}, \{(1,1)\}\}, \\ &\{\{(1,0),(0,1)\} \times \{(1,0),(0,1)\}\}, \\ &\{\{(1,0),(0,1)\} \times \{(0,1),(1,0),(1,1)\}\}, \\ &\{\{(1,0),(0,1),(1,1)\} \times \{(0,1),(1,0),(1,1)\}\}, \\ &\{\{(1,0),(0,1),(1,1)\}\ )\end{aligned} \tag{16}$$

In (16) the Cartesian products are not expanded in order to retain the readability. The expansion yields 4 pairs of multi-indexes in the row standing for $x \cdot x$, 6 pairs in the row standing for $x \cdot y$ and 9 pairs in the row standing for $y \cdot y$.

Since G_{10} and G_{01} are needed in the computation of G_{11}, the selection of single and pairs of multi-indexes according to (9) is done three times, for $(1,0)$, $(0,1)$ and $(1,1)$. From these computation steps

$$\begin{aligned}\Pi^{1,0} = &\ (\{\{(1,0)\}\}, \{\{(1,0)\}\}, \{\}, \{\}, \{\}, \{\{(1,0)\}\})\\ \Pi^{0,1} = &\ (\{\{(0,1)\}\}, \{\{(0,1)\}\}, \{\}, \{\}, \{\}, \{\{(0,1)\}\})\\ \Pi^{1,1} = &\ (\{\}, \{\{(1,1)\}\}, \{\{(1,0),(0,1)\}, \{(0,1),(1,0)\}\},\\ &\ \{\{(1,0),(0,1)\}, \{(0,1),(1,0)\}\},\\ &\ \{\{(1,0),(0,1)\}, \{(0,1),(1,0)\}\}, \{\{(1,1)\}\})\end{aligned} \tag{17}$$

is obtained. The initial number of 27 single and pairs of multi-indexes in (17) has been reduced to 8 in $\Pi^{1,1}$ of (18). From (18) the equations to be solved for G_{10}, G_{01} and G_{11} are constructed by applying (12), adding the terms in the sets and performing the scalar product (10). The results are:

$$\begin{aligned}G_{10}: &\quad a_1 - a_1 G_{10} - C\lambda_1 G_{10} = 0\\ G_{01}: &\quad a_1 - a_1 G_{01} - C\lambda_2 G_{01} = 0\\ G_{11}: &\quad -a_1 G_{11} - C(\lambda_1 + \lambda_2) G_{11} + 2a_2\\ &\quad -2a_2 G_{10} - 2a_2 G_{01} + 2a_2 G_{10} G_{01} = 0\end{aligned} \tag{18}$$

The solutions of the above equations are

$$\begin{aligned}G_{10} = &\ \frac{a_1}{a_1 + C\lambda_1}, \quad G_{01} = \frac{a_1}{a_1 + C\lambda_2}\\ G_{11} = &\ \frac{2a2 - 2a_2 G_{01} - 2a_2 G_{10} + 2a_2 G_{10} G_{01}}{a_1 + C(\lambda_1 + \lambda_2)}\end{aligned} \tag{19}$$

Using $H_N = G_{11..1}/N!$ and replacing G_{01} and G_{10} in (19) yields the desired 2nd order Volterra Kernel $\underline{H}_2 = \underline{H}_2(\omega_1, \omega_2)$ of the system (13):

$$\underline{H}_2 = \frac{-a_2 C^2 \omega_1 \omega_2}{(a_1 + \mathrm{j}\,\omega_1 C)(a_1 + \mathrm{j}\,\omega_2 C)(a_1 + \mathrm{j}\,(\omega_1 + \omega_2) C)} \tag{20}$$

2.2. Multi-index Algorithm for VNVA

The multi-index algorithm for the Volterra nodal voltage analysis, described for instance in [WS98] is carried through in a very similar way. In this method linear systems of equations of the form

$$\boldsymbol{K}^{\mathbf{1}}(s_1 + \ldots + s_n, \boldsymbol{p}) \cdot \underline{\mathbf{H}}_n(s_1, \ldots, s_n) = \underline{\mathbf{I}}^n_{\mathcal{P}}(s_1, \ldots, s_n) \tag{21}$$

are constructed and solved in order to obtain the nth order Volterra kernels $\underline{\mathbf{H}}_{n,m}(s_1, \ldots, s_n)$ of the nodal voltages m. $\boldsymbol{K}^{\mathbf{1}}(s_1 + \ldots + s_n, \boldsymbol{p})$ is the conductance matrix of the nodal voltage analysis of the linearized part of the circuit in the transform domain, with the parameters $\boldsymbol{p}$ of the circuit elements. It is structurally unchanged in the algorithm but its parameterization by $s_1 + \ldots + s_n$ changes with the order n of the Volterra kernels to be computed.

The structure of the right hand side of (21), $\underline{\mathbf{I}}^n_{\mathcal{P}}(s_1, \ldots, s_n)$, depends on the polynomial nonlinearities of the circuit. $\underline{\mathbf{I}}^n_{\mathcal{P}}$ is decomposed in the same way given in (10). The nodal voltages v_m in the products in the resulting vectors Π^n are replaced by the lower order Volterra kernels $\underline{H}_{<n,m}$ of the nodal voltage v_m. Only products, where every $s_1, \ldots, s_n$ appears exactly once as a parameter of the lower order Volterra kernels $\underline{H}_{<n,m}$ contribute to the computation of $\underline{\mathbf{H}}_n$.

Thus, the problem of constructing $\underline{\mathbf{I}}^n_{\mathcal{P}}(s_1, \ldots, s_n)$ is very similar to the construction of the equations for $G_{\boldsymbol{m}^{N,k}}$ in the GEM. It can be efficiently solved without tedious formula manipulations and generation of unnecessary terms by applying multi-index computations here.

3. Conclusions

A class of multi-index algorithms for the efficient computation and application of Volterra kernels for circuits and systems has been developed. The algorithms were implemented in a demonstration program system called PaNAMa. PaNAMa combines C-routines for the communication and the simulation and Mathematica-packages for the computations of the Volterra kernels using the described multi-index algorithms. Details of described algorithms can be found in [Bau04]. The presented ideas could be the key to a very efficient and reliable toolbox for the nonlinear small signal analysis of weakly nonlinear circuits and systems based on Volterra series. Professional programming would even allow to implement the algorithms without using computer algebra such as Mathematica.

References

[Bau04] Andreas Bauer. *Analysis algorithms for nonlinear circuits and systems based on Volterra series (in german).* PhD thesis, TU Dresden, 2004.

[CN79a] L.O. Chua and C.Y. Ng. Frequency domain analysis of nonlinear systems: formulation of transfer functions. *Electronic Circuits and Systems*, 3(6), November 1979.

[CN79b] L.O. Chua and C.Y. Ng. Frequency domain analysis of nonlinear systems: general theory. *Electronic Circuits and Systems*, 3(4), July 1979.

[Rug81] Wilson J. Rugh. *Nonlinear System Theory, The Volterra/Wiener Approach.* The Johns Hopkins University Press, 1981.

[WS98] Piet Wambacq and Willy Sansen. *Distortion Analysis of Analog Integrated Circuits.* Kluwer Academic Publishers, 1998.