Volterra black-box model of electron devices nonlinear behavior based on Neural Network parameters

GEORGINA STEGMAYER
MARCO PIROLA
GIOVANNI GHIONE
Electronics Department
Politecnico di Torino
Cso. Duca degli Abruzzi 24
10129 Torino
ITALY

GIANCARLO ORENGO
PAOLO COLANTONIO
Electronics Engineer Department
Università di Roma II
via Politecnico 1
00133 Roma
ITALY

Abstract. With this paper we want to present a black-box model, that can be applied to a vast number of RF electron devices (e.g. FET). We will show that an analytical Volterra series approximation of the nonlinear behavior time-dependent model of an electron device can be built using a neural network and its parameters, once the proper training data are given.

Key-Words: black-box model, nonlinearity, Volterra model, Volterra Kernels, neural network parameters.

1 Introduction

The classical representation of nonlinearities inside an electronic device/element are usually modeled thorough equivalent circuit, assigning the constitutive relations among the current or charge-voltage relationships and the controlling voltages. This procedure is based on the known physical behavior of the modeled device that dictates the model topology. For this reason not only the model must be tailored to the device, but also the extraction of its parameters strongly depends on it.

Other approaches claim to represent a general device and are referred to as *black-box* model, but that it is generally only partially true, since some assumption on the device is always done. The Volterra expansion approach has been proposed since many years, at least in principle, for non-linear modeling, through an input-output relationship. Due to the difficulties in identifying the higher order kernels through experimental data, this approach is complex. For this reason it is not effectively used within commercially available circuit simulator, despite the fact that it can originate intrinsically a black-box model and, therefore, device independent model.

On the other side the neural network approach to device modeling has received increasing attention in recent years since model tailoring to the device under study only needs a training procedure based on experimental data.

In this paper we propose a black-box time dependent neural network model able to build a Volterra series approximation suitable for electronic devices, allowing its kernels evaluation as well. In this paper we will show, as a case of study, the results obtained using a Curtice MESFET model [1] as the reference drain current.

The Volterra series approach is explained in Section 2. Our proposed dynamic neural network model appears in Section 3. Simulation results are presented in Section 4 and the conclusions of this work can be found on Section 5.

2 Volterra series model of nonlinear behavior

A non-linear dynamical system can be represented exactly by a converging infinite series of the form (1), that reports the dynamic expansion of a single-input single-output system. This equation is known as the

Volterra series expansion [2]. The functions h_0, h_1, h_2,…,h_n are known as the Volterra *kernels* of the system. In general, h_n is the n^{th} order kernel of the series that completely characterizes the n^{th} order nonlinearity of the system. If the continuous Volterra series model (1) is express in discrete form, then it becomes (2). In practice, the Volterra series must be simplified to avoid summations over an infinite number of terms. A sufficiently accurate model can be obtained by using a finite number of term and less than infinite memory [3].

$$y(t) = h_0 + \int_{t=0}^{\infty} h_1(t)x(t-t)dt + \tag{1}$$

$$\int_{t_1=0}^{\infty}\int_{t_2=0}^{\infty} h_2(t_1,t_2)x(t-t_1)x(t-t_2)dt_1 dt_2 + \ldots$$

$$y(k) = h_0 + \sum_{n=0}^{\infty} h_1(n)x(k-n) + \tag{2}$$

$$\sum_{n_1=0}^{\infty}\sum_{n_2=0}^{\infty} h_2(n_1,n_2)x(k-n_1)x(k-n_2) + \ldots$$

The Volterra series analysis is well suited to the simulation of nonlinear microwave devices and circuits, in particular in the weakly nonlinear regime where a few number of kernels are able to capture the device behavior (e.g. for PA distortion analysis). The Volterra kernels allow the analysis of device characteristics of great concern for the microwave designer, such as harmonic generation and inter-modulation phenomena in the case of a FET [4].

The number of terms in the kernels of the series increases exponentially with the order of the kernel and the delay (k) used . This is the most difficult problem with the Volterra series approach and imposes restrictions on its application to many practical systems, which are restricted to use second order models because of the difficulty in kernels expression. A common approach for kernels identification is the use of Wiener orthogonal functionals, but the problem of number of terms in the functionals is still present [5]. Moreover, the use of alternative methods for kernels identification [6], analytical expression [7] or measurement [8] can be a complex and time-consuming task.

In the Biology field, these authors [9][10] outline a method for extracting the Volterra kernels as a function of the weights and bias values of a neural network. Based on this idea, other works propose different strategies for the kernels calculation with different neural networks topologies [11] [12] [13]. However, all of these approaches use a discrete temporal behavior by means of multi-delayed samples of a unique input variable. Our proposed model, instead, is more general in the sense that allows to represent not only the time-domain dependence on one variable, but also a function depending on two or more variables.

Differently from [14], where different neural networks models are used to describe the current nonlinearity using measurements of the current and all its derivatives, our proposed modeling approach [15] (extended here to a general case), instead, is very simple and straightforward, and only needs input/output measurements for the training of one standard MLP model. With only those elements, after performing some very simple calculations, the Volterra series and its kernels can be obtained.

3 Neural Network based approach

The topology of the Multi-layer Perceptron (MLP) neural network that we propose is a very simple one. The first layer, as shown in Fig. 1, has as inputs the time samples of independent variables together with their delayed values. In our case, we have two input variables, x_a and x_b, which correspond, in a MESFET case, to the control voltages *Vds* and *Vgs*, respectively. In the hidden layer there are as many hidden neurons as input neurons, with the hyperbolic tangent as activation function. They receive the sum of the weighted inputs plus a corresponding bias value b_k.

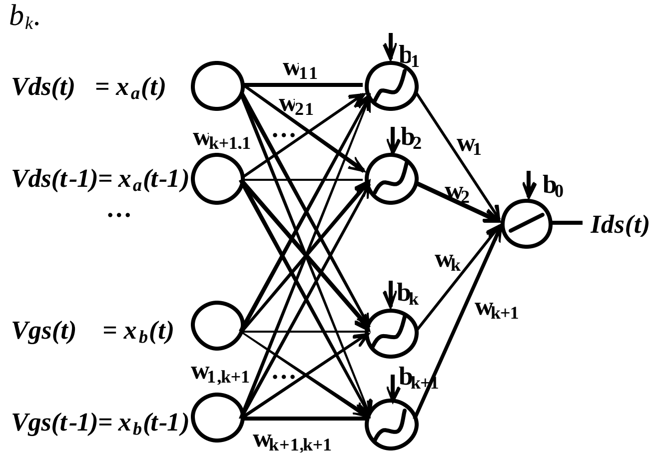

Fig. 1. Dynamic Neural Network model

The output of the neural network model (3) (assuming the output is the drain current of a FET) is calculated as the sum of the weighted outputs of the hidden neurons plus a bias. The activation function of the output neuron is chosen to be linear.

$$Ids(t) = b_0 + \sum_{i=1}^{\infty} w_i \tanh(b_i + \sum_{j=0}^{\infty} w_{i,j+1} x_l (t - j)) \quad (3)$$

where l=[a,b]

Following the approach in [9], we expand the output of our network model (3) as a Taylor series around the bias values of the hidden nodes (4) (where $tanh^{(j)}$ is the j^{th} derivative of the hyperbolic tangent).

$$Ids(t) = b_0 + \sum_{i=1}^{\infty} w_i \sum_{j=0}^{\infty} \left[\frac{\tanh^{(j)}(b_i)}{j!} \left(\sum_{k=1}^{\infty} w_{i,k} x_l (t-(k-1)) \right)^j \right] \quad (4)$$

where l=[a,b]

Accommodating the terms according to their derivative order, yields (5).

$$\begin{aligned}
Ids(t) = {} & b_0 + \sum_{i=1}^{\infty} w_i \tanh(b_i) + \quad (5) \\
& \left[\sum_{j=1}^{\infty} w_j w_{j1} (\tanh^{(1)}(b_j)) \right] x_a(t) + \\
& \left[\sum_{j=1}^{\infty} w_j w_{j2} (\tanh^{(1)}(b_j)) \right] x_a(t-1) + \ldots + \\
& \left[\sum_{j=1}^{\infty} w_j w_{jk} (\tanh^{(1)}(b_j)) \right] x_b(t) + \\
& \left[\sum_{j=1}^{\infty} w_j w_{jk+1} (\tanh^{(1)}(b_j)) \right] x_b(t-1) + \ldots + \\
& \left[\sum_{i=1}^{\infty} \sum_{j=1}^{\infty} w_i w_{j1} w_{j1} \left(\frac{\tanh^{(2)}(b_i)}{2} \right) \right] x_a(t)^2 + \\
& \left[\sum_{i=1}^{\infty} \sum_{j=1}^{\infty} w_i w_{j2} w_{j2} \left(\frac{\tanh^{(2)}(b_i)}{2} \right) \right] x_a(t-1)^2 + \ldots + \\
& \left[\sum_{i=1}^{\infty} \sum_{j=1}^{\infty} w_i w_{jk} w_{jk} \left(\frac{\tanh^{(2)}(b_i)}{2} \right) \right] x_b(t)^2 + \\
& \left[\sum_{i=1}^{\infty} \sum_{j=1}^{\infty} w_i w_{jk+1} w_{jk+1} \left(\frac{\tanh^{(2)}(b_i)}{2} \right) \right] x_b(t-1)^2 + \ldots + \\
& \left[\sum_{i=1}^{\infty} \sum_{j=1}^{1} \sum_{k=k}^{k} w_i w_{ij} w_{ik} \left(\frac{\tanh^{(2)}(b_i)}{2} \right) \right] x_a(t) x_b(t) + \ldots
\end{aligned}$$

Considering now i.e., only the terms that contain the variable $x_a(t)$, it is easy to recognize the terms between brackets in (5) as the Volterra kernels of a Volterra series expansion for the relationship $Ids(x_a(t))$. Extending the Volterra series approximation to more than one variable, we can see now how simple is to use the Neural Network parameters to obtain the Volterra kernels values for a multivariable case as well.

4 Case of study and Simulation results

To clarify our proposed approach and as an initial "case of study", we will present here a simple example of a function that depends on two variables, and for the analysis we will use a simplified version of the neural network model presented in Fig. 1.

We compare the expression (6), that reports a two variables (*Vds*, *Vgs*) function *Ids* approximated through its polynomial Volterra series expansion up to the 3rd order, with the terms in (5). The coefficients of the series are the first order (*Gds* and *Gm*), second order (*Gds2*, *Gm2* and *Gmd*) and third order (*Gds3*, *Gm3*, *Gm2d*, *Gmd2*) derivatives of the current with respect to the voltages. These coefficients happen to be the Volterra kernels of the series. If we consider $Vds = x_a$ and $Vgs = x_b$, it is straightforward to recognize the terms between brackets in (5) as the Volterra kernels of the series (6) for this static case.

$$\begin{aligned}
Ids(Vds, Vgs) = {} & Ids(Vds0, Vgs0) + \quad (6) \\
& Gds \cdot vds + Gm \cdot vgs + \\
& Gds2 \cdot vds^2 + Gm2 \cdot vgs^2 + \\
& Gmd \cdot vgs \cdot vds + \\
& Gds3 \cdot vds^3 + Gm3 \cdot vgs^3 + \\
& Gm2d \cdot vgs^2 \cdot vds + Gmd2 \cdot vgs \cdot vds^2
\end{aligned}$$

We use a simplified version of our original neural network model in Fig.1 (two input nodes, two hidden nodes, one output node, see Fig. 2), while concerning the data for the training phase we use data obtained from a cubic Curtice model (A_0=0.0625, A_1=0.05, A_2=0.01, A_3=0.001). The network has been trained with 200 samples uniformly distributed. Although the Back-Propagation algorithm is generally used for the training of MLP networks, its convergence parameters have to be finely adjusted to get fast convergence. That is why in the simulations in this paper we have used the Levenberg-Marquardt algorithm to train the weights and bias values of the networks, which provides relatively fast training and no adjusting of step and momentum terms are required to obtain convergence. The training has been refined up to an average relative error of 1e-07.

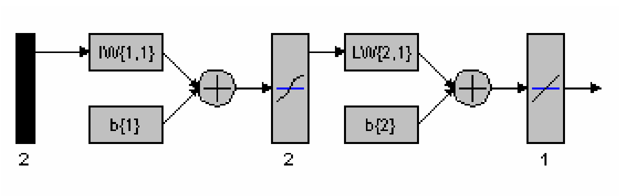

Fig. 2. Simplified Neural Network model

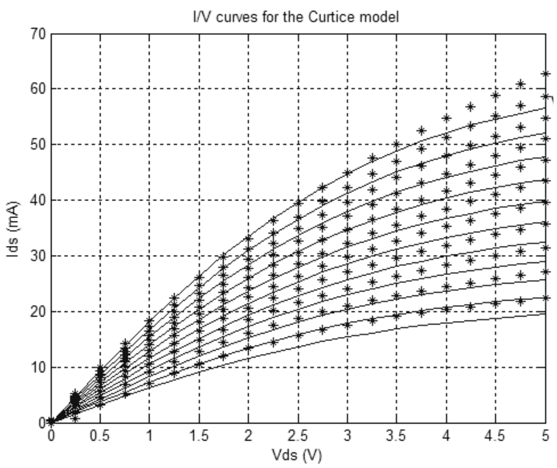

Fig. 3. Curtice model I/V Curves (-) vs. Volterra-Neural Network approximation (*).

Volterra kernels	value
$h_1(0)\,(Gds)$	0.02
$h_1(1)\,(Gm)$	0.002
$h_2(0,0)\,(Gds2)$	-0.002
$h_2(1,1)\,(Gm2)$	0.01
$h_2(0,1)\,(Gmd)$	0.01
$h_3(0,0,0)\,(Gds3)$	0.0001
$h_3(1,1,1)\,(Gm3)$	0.01
$h_3(0,0,1)(Gmd2)$	-0.0004
$h_3(1,1,0)(Gm2d)$	0.0003

Table 1. Volterra kernels values

At the end of the training procedure, the Volterra kernels up to the 3rd order have been extracted and compared with the ones analytically computed from the Curtice approximations. The approximation was found very accurate and the kernels values are reported in Table 1. Fig. 3 also reports, for some values of the gate current, the drain current obtained from the Volterra-Neural-Network approximation (*) and the virtual experimental data from the Curtice model (-). Keeping in mind that the Volterra series is an approximation and that here only up to the 3rd order terms have been included in the series, we conclude that the agreement is quite good.

5 Conclusions

We have developed a new Neural Network model that allows the building of a Volterra black-box model and the extraction of its Volterra Kernels in the case of a dynamic system with multiple driving voltages, and so able to reproduce the non-linear device behavior.

We have explained how it is possible to obtain the Volterra series analytical expression for an electronic device nonlinear behavior, which sometimes can be a difficult task, using parameters of a Neural Network model only trained with input/output measurements, and how to calculate its Volterra Kernels. The model has been validated: the results on a case of study (a Curtice FET current) are here reported.

References

[1] W. R. Curtice, M. Ettenberg, 'A Nonlinear GaAs FET Model for Use in the Design of Output Circuits for Power Amplifiers', *IEEE Transactions on Microwave Theory and Technicques*, vol. 33, no. 12, 1985, pp. 1383-1394.

[2] W. Rough, *Nonlinear System Theory. The Volterra/Wiener Approach*, Ed. Johns Hopkins University Press, 1981.

[3] S. Boyd, L. O. Chua, C.A. Desder, 'Analytical Foundations of Volterra Series', *IMA Journal of Mathematical Control & Information*, no. 1, 1984, pp. 243-282.

[4] S. A. Maas, *Nonlinear Microwave Circuits*, Ed. Artech House, 1988.

[5] G. Govind, P.A. Ramamoorthy, 'Multi-layered Neural Networks and Volterra Series: The Missing Link", *IEEE International Conference on Systems Engineering*, vol. 1, 1990, pp. 633-636.

[6] H. Kashiwagi, H. Harada, L. Rong, 'Identification of Volterra kernels of nonlinear systems by separating overlapped kernel slices', *Conference of the Society of Instrument and*

Control Engineers of Japan, no. 2, 2002, pp. 707-712.

[7] Bauer, W. Schwarz, 'Circuit analysis and optimization with automatically derived Volterra kernels', *IEEE International Symposium on Circuits and Systems*, 2000, pp. I-491 - I-494.

[8] S. Boyd, Y.S. Tang, L.O. Chua, 'Measuring Volterra Kernels', *IEEE Transactions on Circuits and Systems*, vol. 30, no. 8, 1983, pp. 571-577.

[9] J. Wray, G.G.R. Green, 'Calculation of the Volterra kernels of non-linear dynamic systems using an artificial neural network', *Biological Cybernetics*, no. 71, 1994, pp. 187-195.

[10] W. P. Woods, A. H. Batchelor, S. R. Johnson, G. G. R. Green, 'Comparison of the response of a time delay NN with an analytical model', *International Joint Conference on Neural Networks*, no. 2, 2002, pp. 1120-1125.

[11] D. I. Soloway, J. T. Bialasiewicz, 'Neural Networks modeling of nonlinear systems based on Volterra series extension of a linear model', *IEEE International Symposium on Intelligent Control*, vol. 1, 1992, pp. 7-12.

[12] V. Z. Marmarelis, X. Zhao, 'Volterra models and Three Layers Perceptron', *IEEE Transactions on Neural Networks*, vol. 8, no. 6, 1997, pp. 1421-1433.

[13] N. Z. Hakim, J. J. Kaufman, G. Cerf, H. E. Meadows, 'Volterra characterization of Neural Networks', *Signals, Systems and Computers*, no. 2, 1991, pp. 1128-1132.

[14] J. A. Garcia, A. Tazon Puente, A. Mediavilla Sanchez, I. Santamaria,M. Lazaro, C. J. Pantaleon, J. C. Pedro, 'Modeling MESFETs and HEMTs Intermodulation Distortion Behavior Using a Generalized Radial Basis Function Network', *International Journal on RF and Microwave Computer-Aided Design, Special number on Neural Networks*, no. 9, 1999, pp. 261-276.

[15] G. Stegmayer, M. Pirola, G. Orengo, O. Chiotti, 'Towards a Volterra series representation from a Neural Network model', 5th *WSEAS International Conference on Neural Networks and Applications*, to be published, 2004.