Towards a Volterra series representation from a Neural Network model

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

GIANCARLO ORENGO
Electronics Engineer Department
Università "Tor Vergata"
via Politecnico 1
00133 Roma
ITALY

OMAR CHIOTTI
G.I.D.S.A.T.D.
Universidad Tecnológica Nacional
Lavaise 610
3000 Santa Fe
ARGENTINA

Abstract. We have developed a Neural Network model able to reproduce some nonlinear characteristics of an electronic device. However, electronic devices nonlinear analysis requires an analytical model, that allows to draw conclusions about the device behavior. Such a model can be the Volterra series representation, which is a series that has some particular terms, named the "Volterra kernels". We want to show in this work how a Volterra model can be built using the parameters of the proposed Neural Network model. We present a method for estimating the Volterra kernels using the Neural Network parameters and some simulation results.

Key-Words: nonlinear behavior, neural networks parameters, Volterra series, Volterra kernels

1 Introduction

The classic representation of a nonlinearity inside an electronic device/element, based on a nonlinear equivalent circuit, is generally an equation, such as current-voltage and charge-voltage relationships. These models, however, only describe the input/output behavior of the device/element. For example, the generally known and widely accepted Field Effect Transistor (FET) model: the Curtice Model [1]. The current-voltage equation for the cubic Curtice model is

$$Ids = (A_0 + A_1V_1 + A_2V_1^2 + A_3V_1^3)\tanh(cVds) \quad (1)$$

where

V_1=Vgs[1 + β(Vds0-Vds),

Vgs and Vds are the intrinsic voltages (gate to source voltage and drain to source voltage, respectively), and Ids is the drain to source current. This model reflects the input/output behavior of the drain to source current Ids, that is function of the intrinsic voltages of the FET, Vgs and Vds.

Simulations performed in a microwave circuits simulator show the I/V (current vs. voltage) Curves of the model (Fig. 1), for different voltages combinations, using the following values: Vgs = [-1…0] step 0.2, Vds = [0…5] step 0.5, β = 0, χ = 0.3.

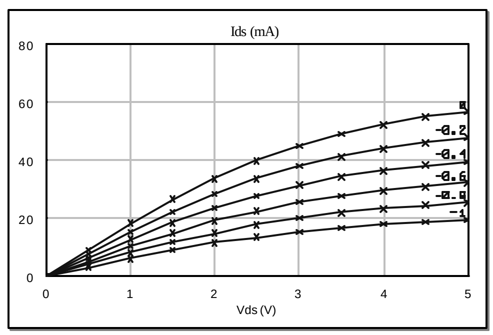

Fig. 1. I/V Curves for the Curtice model

The current Ids accounts for most of the nonlinear behavior in the device. However, there are other elements that also contribute to the nonlinearity of the model: the capacitances. The capacitance relationship in function of the voltage C(V) is given in (2) where C_{J0} is the zero bias capacitance, ϕ is the barrier height (V≤ϕ) and γ is the grading coefficient (an empirical value, 1/3 ≤ γ ≥ ½).

$$C(V) = \frac{C_{J0}}{\left(1 - \frac{V}{f}\right)^{g}} \quad (2)$$

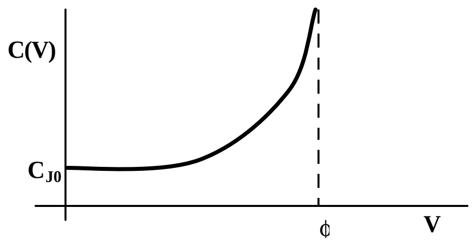

Fig. 2. Theoretical capacitance curve

These models, however, only show the input/output characteristics of the current (1) and the capacitance (2), but do not allow a deeper analysis about the device behavior that could help an electronic designer. One of the approaches regarding this analysis is the approximation of the nonlinear behavior of the element under consideration with a numerical series such as the Volterra series. This series has some terms named "kernels" that allow a deeper understanding of the device. Its main disadvantage, however, is the analytical expression or calculation of its kernels. Our proposal is to use a neural network and its parameters, to help in the building of the Volterra series and the calculation of its kernels.

In section 2 we present the Volterra series analysis and related work regarding the use of neural networks for Volterra kernels calculation. In section 3 we present our Neural Network based model and how to calculate the kernels from parameters of the network. Simulations results are presented in section 4. The conclusions of the work can be found in section 5.

2 Volterra Series

For a single-input, single-output (SISO) non-linear dynamical system, with an output time function, y(t), and an input time function, x(t), it can be represented exactly by a converging infinite series of the form

$$y(t) = \sum_{1}^{\infty} y_n(t) \tag{3}$$

$$y_n(t) = \int_{-\infty}^{\infty} \ldots \int_{-\infty}^{\infty} h_n(t_1 \ldots t_n) \prod_{j=1}^{n} x(t - t_j) dt_j \tag{4}$$

This system can be represented to any desired degree of accuracy by a finite series of the form (5). This equation is known as the Volterra series expansion. The $h_0, h1, h_2, \ldots, h_n$ are known as the Volterra kernels of the system [2] [3]. The kernel h_0 is called the impulse response of the system, h_1 is the first order kernel, h_2 is the second order kernel, and in general, h_n is the nth order kernels of the series.

$$y(t) = h_0 + \int_0^{\infty} h_1(t) x(t - t) dt + \int_0^{\infty} \int_0^{\infty} h_2(t_1, t_2) x(t - t_1) x(t - t_2) dt_1 dt_2 + \ldots \tag{5}$$

If the continuous Volterra series model (4) is express in discrete form, then it becomes

$$y(k) = h_0 + \sum_0^{\infty} h_1(n) x(k - n) + \sum_0^{\infty} \sum_0^{\infty} h_2(n_1, n_2) x(k - n_1) x(k - n_2) + \ldots \tag{6}$$

In nonlinear microwave analysis, in particular for small signal regime, the tool for excellence has been the Volterra-series analysis [4]. The Volterra description for an electronic device is based on Taylor-series expansions of the device nonlinearity around a fixed bias point or around a time-varying signal.

Example of the first type of model is the Ids current in a FET around the bias voltages (7) (in this case the series is developed up to the third order kernels), where Vgs0 andVds0 are the internal bias voltages and vgs and vds are the incremental intrinsic voltages respect to the bias voltages. The coefficients of the series are the first order (Gm1 and Gds), second order (Gm2, Gds2 and Gmd) and third order (Gm3, Gds3, Gm2d, Gmd2) derivatives of the current with respect to the voltage. These coefficients happen to be the Volterra kernels of the series. An example of the second type of Volterra model can be the capacitance C in function of a time varying voltage (8), considering *x(k)* as a discrete sample of the voltage (in this case, we have used two samples, *x(0)=V(t)* and *x(k-1)=V(t-1)*).

$$\begin{aligned} Ids = {} & Ids(Vgs0, Vds0) \\ & + Gm1.vgs + Gds.vds \\ & + Gm2.vgs^2 + Gds2.vds^2 + Gmd.vgs.vds \\ & + Gm3.vgs^3 + Gds3.vds^3 \\ & + Gm2d.vgs^2.vds + Gmd2.vgs.vds^2 \end{aligned} \tag{7}$$

The derivatives allow the inference of some device characteristics of great concern for the microwave designer, i.e. nonlinear distortion phenomena. We can say that a good device model not only must accurately reflect its nonlinear characteristics but also its derivative information. That kind of information can be inferred from the Volterra model and its kernels.

$$\begin{aligned} C = & h_0 + h_1(0)x(k) + h_1(1)x(k-1) \\ & + h_2(0,0)x(k)^2 + h_2(1,1)x(k-1)^2 \\ & + h_2(0,1)x(k)x(k-1) \\ & + h_3(0,0,0)x(k)^3 + h_3(1,1,1)x(k-1)^3 \\ & + h_3(0,0,1)x(k)^2 x(k-1) \\ & + h_3(0,1,1)x(k)x(k-1)^2 \end{aligned} \tag{8}$$

However, kernels calculation, analytical expression or measurement is very a complicated and time-consuming task [5][6][7]. Some authors have proposed a method of extracting the Volterra kernels of any order as a function of the weights and bias values of a feed-forward time delayed neural network with one hidden layer [8][9]. Even kernels calculation with different neural networks topologies have been proposed [9][11][12], also in the electronics field [13]. In [14] it is compared the use of two types of Radial Basis Function neural networks and an MLP to describe the current nonlinearity, where measurements of the current and its derivatives are necessary for the model. All of these approaches deal with time series inputs and only one variable. In our case, however, we want to represent not only that, but also a function that depends on two variables and its model is developed around bias points.

Our approach is very simple and straightforward, only input/output device measurements or simulations are necessary, and the training of a standard MLP Neural Network model. With only that elements, after performing some very simple calculations, the Volterra series and its kernels can be obtained, for any type of Volterra representation.

3 Proposed Neural Network model

The topology of the neural network that we propose is a very simple one (Fig. 3). The input layer has two inputs (a and b), which are multiplied by its corresponding weights w_{ij} (i,j=[1…2]) and propagated through the network. In the hidden layer there are two hidden neurons. As their activation function, following [8], we have chosen the hyperbolic tangent (tanh). Each of them receives the sum of the weighted inputs plus its corresponding bias value b_k (k=[1…2]). The output neuron has a linear activation function and therefore the output of the neural network (y) is calculated as the sum of the weighted outputs of the two hidden neurons plus a bias (9).

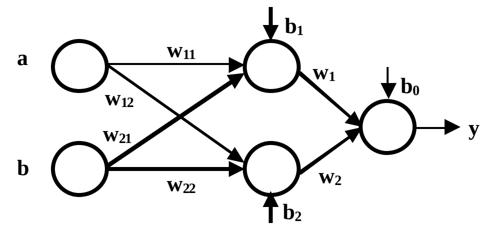

Fig. 3. Proposed Neural Network model

$$\begin{aligned} y = & b_0 + \\ & w_1[\tanh(b_1 + w_{11}a + w_{21}b)] + \\ & w_2[\tanh(b_2 + w_{12}a + w_{22}b)] \end{aligned} \tag{9}$$

We want to show the equivalence between the output of our neural network model (9), that will learn the nonlinear relationship of some element in the Curtice model, and its Volterra series representation ((7) for the current, (8) for the capacitance).

Following the approach in [8], we expand the output of our network model (9) as a Taylor series around the bias values of the hidden nodes

$$y = b_0 + \sum_{i=1}^{2} w_i \sum_{j=0}^{\infty} \left[\frac{\tanh^{(j)}(b_i)}{j!} \left(\sum_{i=1}^{2} w_{ij+1}a + w_{j+1i}b \right)^j \right] \tag{10}$$

where $tanh^{(j)}$ is the j[th] derivate of the hyperbolic tangent (*tanh*). Developing the brackets in (10) and accommodating the terms according to their derivative order, yields (11).

In the case of the current Ids (7), considering a = vgs, b = vds and y = Ids, it is quite straightforward to recognize the kernels as the terms between brackets in (11). It is quite simple also to calculate the values of these kernels using the neural network parameters: the weights, the bias values and the derivatives of the hyperbolic tangent. Similarly, with the Capacitance model (8), considering $a = V(t) = 0$, $b = V(t-1) = 1$ and

y = C, here are also evident the Volterra kernels, and how they can be calculated from the neural network parameters.

$$y = b_0 + \sum_{i=1}^{2} w_i \tanh(b_i) + \tag{11}$$
$$\left[w_1 w_{11} (\tanh^{(1)}(b_1)) + w_2 w_{12} (\tanh^{(1)}(b_2))\right] a +$$
$$\left[w_1 w_{21} (\tanh^{(1)}(b_1)) + w_2 w_{22} (\tanh^{(1)}(b_2))\right] b +$$
$$\left[\left(\frac{\tanh^{(2)}(b_1)}{2}\right) w_1 w_{11} w_{11} + \left(\frac{\tanh^{(2)}(b_2)}{2}\right) w_2 w_{12} w_{12}\right] a^2 +$$
$$\left[\left(\frac{\tanh^{(2)}(b_1)}{2}\right) w_1 w_{21} w_{21} + \left(\frac{\tanh^{(2)}(b_2)}{2}\right) w_2 w_{22} w_{22}\right] b^2 +$$
$$\left[\left(\frac{\tanh^{(2)}(b_1)}{2}\right) w_1 w_{11} w_{21} + \left(\frac{\tanh^{(2)}(b_2)}{2}\right) w_2 w_{12} w_{22}\right] ab + \ldots$$

Equations (12) to (14) show how to calculate the kernels from the weights and bias values of the network. These are general formulas that allow to calculate any kernel having into account any number of input neurons and any number of hidden neurons. Due to space restrictions we only show the calculations for the impulse response, the first and second order kernels of our proposed neural network model, but also higher order kernels could be obtained with little effort. The hyperbolic tangent derivatives have been developed in the formulas.

$$h_0 = b_0 + \sum_{i=1}^{2} w_i \tanh(b_i) \tag{12}$$

$$h_1(k) = \sum_{j=1}^{2} w_j w_{k+1j} (1 - \tanh^2(b_j)) \tag{13}$$

$$h_2(k_1, k_2) = \frac{\sum_{l=1}^{2} w_j w_{k_1+1j} w_{k_2+1j} (-2\tanh(b_j) + 2\tanh^3(b_j))}{2} \tag{14}$$

where k, k_1, k_2 = [0…1] for the neural network in Fig. 3.

4 Simulation Results

We have used (1) and (2) to generate the training data, simulations performed with an electronics circuits analyzer, but also laboratory measurements could have been used because only the input/output data is necessary.

Once the neural network model was trained, using back-propagation and the Levenberg-Marquardt algorithm, to reproduce the nonlinear behavior of the system, we have extracted the weights and bias values from the neural network topology and have calculated the Volterra kernels of the system up to the third order. Then we have built the functions (7) and (8) with the calculated values, and we have plotted it against the original element behavior. The comparison between the drain to source current Ids original equation (*) and the Volterra model approximation based on the parameters of the neural network (-) can be seen in Fig. 4. It can be seen that our neural network based approximation is very close to the original equation, even considering that the Volterra series is an approximation (the average error is in the order of 1e-07), and therefore the inclusion in the series of more kernels would allow for a better representation. We have also simulated the capacitance curve in function of a voltage following a quadratic law over time. We have set the parameters in equation (2) to some typical values: C_{J0}=1, ϕ=0.7, γ=1/3.

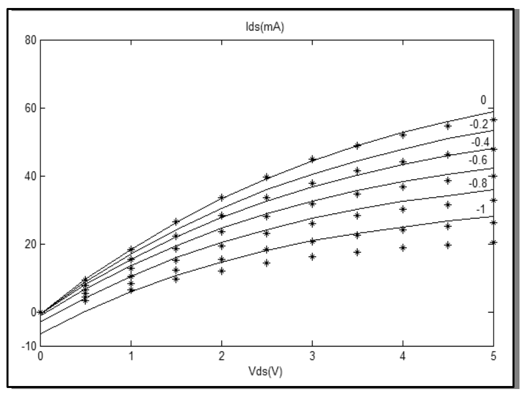

Fig. 4. I/V Curves of the Curtice model (*) vs. I/V Curves of the Volterra-NN model (-)

Once the network was trained, with and average error of 1e-05, we have build different Volterra approximations (8) to be able to compare among them and with respect to the original function C(V). The series containing kernels up to the first order only was called Vh_1. The successive series having higher order kernels (second order, third order) were named respectively Vh_2 and Vh_3 (Fig. 5).

As expected, when more terms are added to the Volterra approximation, it can better reproduce the original function it tries to estimate. We have also performed other simulations slightly changing the topology of the network on Fig. 3 adding one more delayed inputs to the input layer, to see if the kernels obtained were better to calculate the nonlinearity of the system. That is to say, we wanted to check if adding memory to the system (more past values) the approximation obtained would improve. The results are presented in Fig. 6. The Volterra second order approximations with two delayed inputs (V22-h2) and three delayed inputs (V32-h_2) can hardly be distinguished. They have almost the same values. That gives us insight that adding memory to the system does not improve its Volterra approximation, but adding more kernels to the series does.

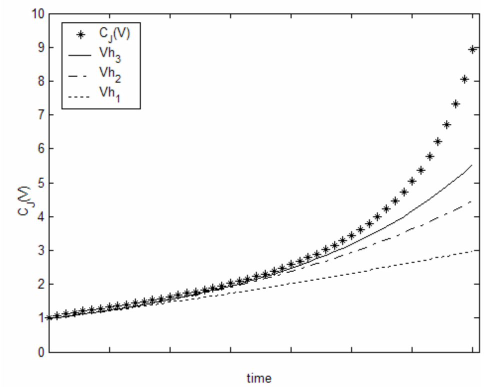

Fig. 5. Results for different approximations

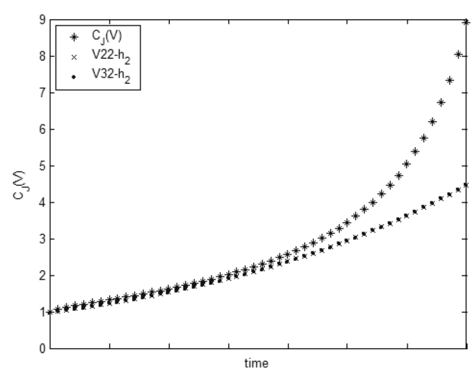

Fig. 6. Comparison of second order kernels

5 Conclusions

We have developed a very simple Neural Network model which can reproduce the nonlinear behavior of an electronic device, in particular a Field-Effect Transistor (FET), modeled with the Curtice equations. The nonlinear elements that we have analyzed for this device were the drain to source current Ids and the capacitance C, using simulation data of its original input/output behavior. However, the construction of another model, the Volterra series model, allows a better understanding of the device non-linearity. In this work we have shown how these two models are related and we have explained here how it is possible to build a Volterra series analytical expression for any nonlinear element, which can be a difficult task, using parameters of a trained neural network model

From our simulation results we conclude that our approach is valid and that even when we have built a series having into account up to the third order kernels only, the approximation of the original function is quite accurate and fast. The simplicity of the calculations allowed us to try different configurations for the neural network to analyze what happens if it is complicated with more input samples, that is to say more memory. Due to the kernels complex calculus, the Volterra series approximations generally are made up to the third order kernels. With the help of the neural networks it could be easy to calculate higher order kernels and therefore obtain a more accurate representation of a nonlinear system. Especially in the case were the system equations are not available but only simulations or input/output measurement data.

We want to highlight that our proposed approach implies and effective and concrete application of a neural network model in the electronics field, that allows the building of an analytical Volterra series representation for any electronic device or element, with the help of a very simple neural network model that needs few data and some algebra, saving precious time to the microwave engineer at the moment of device analysis and design.

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] 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.

[6] A. 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

[12] 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.

[13] Y. Harkouss, J. Rousset, H. Chehade, E. Ngoya, D. Barataud, J. P. Teyssier, ‘Modeling Microwave Devices and circuits for telecommunications systems design’, *International Joint conference on Neural Networks*, vol. 1, 1998, pp. 128-133.

[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.