The Volterra representation of an electronic device using the Neural Network parameters

Georgina S. Stegmayer
Politecnico di Torino, Dipartimento Elettronica,
Torino, Italia, 10129
georgina.stegmayer@polito.it

and

Omar Chiotti
Universidad Tecnologica Nacional, GIDSATD,
Santa Fe, Argentina, 3000
chiotti@ceride.gov.ar

Abstract

Many electronic devices present nonlinear characteristics, which are often difficult to express analytically. Generally it is easier to perform measurements of the device parameters than to develop an analytical model of its behavior. As Neural Networks can be used to learn a system dynamics from input-output data only, we have developed a Neural Network model which reproduces the nonlinear behavior of an electronic device, in particular a Field-Effect Transistor (FET), using simulation data. However, electronic devices nonlinear analysis requires an analytical model (i.e. an equation representing the current-voltage relationship), described as a closed-form function, that allows to draw conclusions about the device, such as the Volterra series model. In this work, we want to show how the neural model and the analytical Volterra series model of the transistor are totally equivalent. Therefore, we show here how it is possible to build an analytical expression for a device nonlinearity, the Volterra series, with parameters of a standard Neural Network, trained with the device measurements or simulation data.
Keywords: Neural Networks, nonlinear electronic devices, Volterra model.

1. INTRODUCTION

Electronic device models based on nonlinear equivalent circuit for commercial CAD programs, generally require current-voltage and charge-voltage mathematical models, described as a closed-form function of the intrinsic control voltages, to characterize the nonlinear circuit elements. This equation describes the input/output behavior of the device/element. One example of such a function is a generally known and widely accepted transistor model: the Curtice Model [3].

One of the elements described by the Curtice Model is the nonlinear characteristic of the current in a Field-Effect Transistor (FET) device. The Curtice model equivalent circuit is presented in Figure 1. 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*. This element accounts for most of the nonlinear behavior of the device. We have performed some simulations to show the I/V (current vs. voltage) curves of the model (Figure 2), for different voltages combinations, using the following values: Vgs = [-1…0], Vds = [0…5], $\beta = 0$, $\chi = 0.3$.

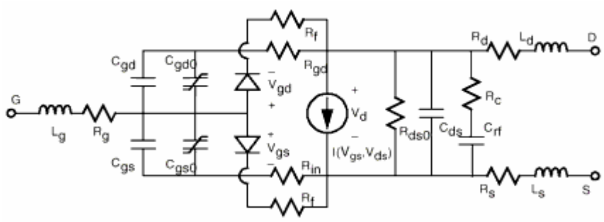

Figure 1. Curtice FET model equivalent circuit

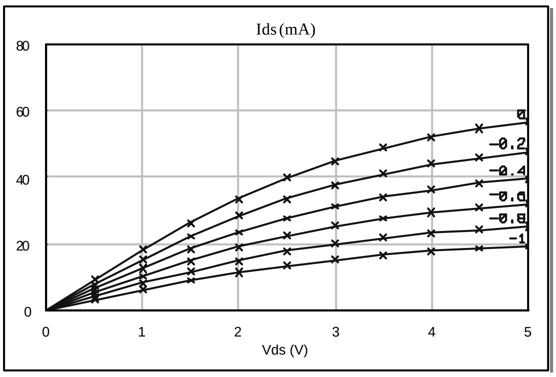

Figure 2. I/V Curves for the Curtice model

The model only shows the input/output characteristic of the element, but does not allow doing 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 [4][2]. 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 that confirm our proposal are shown in Section 4. The conclusions of the work can be found in Section 5.

2. VOLTERRA SERIES ANALYSIS

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

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

$$y_n(t) = \int_{-\infty}^{\infty} \ldots \int_{-\infty}^{\infty} h_n(t_1 \ldots t_n) \prod_{j=1}^{n} x(t - t_j) dt_j \tag{2}$$

This system can be represented to any desired degree of accuracy by a finite series of the form (3). This equation is known as the Volterra series expansion. The $h_1, h_2, \ldots, h_n$ are known as the Volterra *kernels* of the system [8]. 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 n^th order kernel of the series. The Fourier transform of the kernel functions yields the transfer functions of the system. The transform of h_1 gives the linear transfer function and the transforms of the higher order kernels give the non-linear transfer functions.

$$y(t) = h_0 + \int_{t=0}^{\infty} h_1(t)x(t-t)dt + \int_{t_1=0}^{\infty}\int_{t_2=0}^{\infty} h_2(t_1,t_2)x(t-t_1)x(t-t_2)dt_1dt_2 + \ldots + \int_{t_1=0}^{\infty}\ldots\int_{t_n=0}^{\infty} h_n(t_1,t_2,\ldots,t_n)x(t-t_1)x(t-t_2)\ldots x(t-t_n)dt_1dt_2\ldots dt_n + \ldots \quad (3)$$

If the continuous Volterra series model (3) is discretized, then it assumes the form

$$y(k) = h_0 + \sum_{n=0}^{\infty} h_1(n)x(k-n) + \sum_{n_1=0}^{\infty}\sum_{n_2=0}^{\infty} h_2(n_1,n_2)x(k-n_1)x(k-n_2) + \ldots + \sum_{n_1=0}^{\infty}\ldots\sum_{n_n=0}^{\infty} h_n(n_1,n_2,\ldots,n_n)x(k-n_1)x(k-n_2)\ldots x(k-n_n) + \ldots \quad (4)$$

In nonlinear microwave analysis, in particular for small signal regime, the tool for excellence has been the Volterra-series analysis [8]. The Volterra description for an electronic device is based on Taylor-series expansions of the device nonlinearity around a fixed bias point, in the case of amplifiers or transistors, or around a time-varying waveform signal in the case of mixers. The nonlinear behavior in the device represented by the circuit of Figure 1, is particularly due to the drain to source current *Ids*. Its Volterra series expression truncated at the third order kernel is the following

$$\begin{aligned} Ids(Vgs,Vds) = {} & Ids(Vgs0,Vds0) + \\ & Gm1.vgs + Gds.vds + \\ & Gm2.vgs^2 + Gds2.vds^2 + Gmd.vgs.vds + \\ & Gm3.vgs^3 + Gds3.vds^3 + Gm2d.vgs^2.vds + Gmd2.vgs.vds^2 \end{aligned} \quad (5)$$

where *Vgs0* and *Vds0* are the internal bias voltages, *vgs* = *Vgs* - *Vgs0* and *vds* = *Vds* - *Vds0* 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. If we compare (4) and (5) we can see that these coefficients happen to be the Volterra kernels of the series.

These derivatives allow the inference of some device characteristics of great concern for the microwave designer, such as harmonic generation and inter-modulation phenomena in the case of a FET transistor, information that can be inferred from the derivatives of the model, which are contained in the Volterra kernels. For example, the coefficient *Gm1* is the first derivative of the current *Ids* with respect to the voltage *vgs*. It is an important parameter of the device called *transconductance*, which expresses the performance of the transistor

However, kernels calculation, measurement or analytical expression is a very complicated and time-consuming task [7] [1]. In the Biology field, these authors [13][12] have outlined a method for 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. Based on this idea, other works propose different strategies for kernels calculation with different neural networks topologies [11][9], also in the electronics field [5] [6]. But all of these approaches deal with time series inputs and only one variable.

However, in the case we are interested in, the function depends on two variables and the Volterra model is developed around two bias points, therefore the existing models cannot be used. We will propose a new model which can be used also in a multivariable case.

We have chosen to use a feed-forward multilayer perceptron (MLP) neural network, having the two intrinsic voltages as inputs, and the current as output. Differently from [4] where 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 all its derivatives are necessary for the model, which is a complex task; our approach, instead, only needs simple device measurements or simulations of the voltages and the output current, and the training of a very simple MLP neural network model. With only those elements, after performing some very simple calculation, the Volterra series and its kernels can be obtained.

3. Neural Network Based Model

The topology of the MLP network that we propose is a very simple one. It has one input layer, one hidden layer with two hidden neurons, and an output layer with a single output (Figure 3).

The input layer has two neurons, which are the controlled voltages *vgs* and *vds*. The inputs 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 we have chosen the hyperbolic tangent (*tanh*). Each of them receives the sum of the weighted inputs, added to its corresponding bias value b_k (k=[1…2]). The output neuron has a linear activation function and therefore the output of the network *Ids* is calculated as the sum of the weighted outputs of the two hidden neurons plus a bias (6).

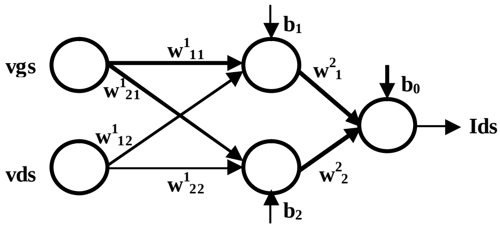

Figure 3. Topology of the MLP studied

$$Ids = b_0 + w_1^2\,[tanh(b_1 + w_{11}^1 vgs + w_{12}^1 vds)] + w_2^2\,[tanh(b_2 + w_{21}^1 vgs + w_{22}^1 vds)] \tag{6}$$

We want to show the equivalence between the output of our neural network model (6), that learns the relationship between the voltages and the current of the Curtice model, and its Volterra series representation (5), and we will show how easy is to obtain, from the network parameters, the values of the Volterra kernels that form the series.

Following the approach in [13], we expand the output of our network model (6) as a Taylor series around the bias values of the hidden nodes

$$Ids = b_0 + w_1^2 \sum_{j=0}^{\infty}\left[\frac{tanh^{(j)}(b_i)}{j!}\left(w_{11}^1 vgs + w_{12}^1 vds\right)^j\right] + w_2^2 \sum_{j=0}^{\infty}\left[\frac{tanh^{(j)}(b_i)}{j!}\left(w_{21}^1 vgs + w_{22}^1 vds\right)^j\right] \tag{7}$$

where $tanh^{(j)}$ is the *jth* derivate of the hyperbolic tangent (*tanh*).

Developing the brackets in (7) up to the second order derivatives and accommodating the common terms, yields

$$\begin{aligned} Ids = {} & b_0 + \sum_{i=1}^{2} w_i^2\, tanh(b_i) + \\ & \left[w_1^2 w_{11}^1 (tanh^{(1)}(b_1)) + w_2^2 w_{21}^1 (tanh^{(1)}(b_2))\right] vgs + \\ & \left[w_1^2 w_{12}^1 (tanh^{(1)}(b_1)) + w_2^1 w_{22}^1 (tanh^{(1)}(b_2))\right] vds + \\ & \left[w_1^2 w_{11}^1 w_{11}^1 \left(\frac{tanh^{(2)}(b_1)}{2}\right) + w_2^2 w_{21}^1 w_{21}^1 \left(\frac{tanh^{(2)}(b_2)}{2}\right)\right] vgs^2 + \\ & \left[w_1^2 w_{12}^1 w_{12}^1 \left(\frac{tanh^{(2)}(b_1)}{2}\right) + w_2^2 w_{22}^1 w_{22}^1 \left(\frac{tanh^{(2)}(b_2)}{2}\right)\right] vds^2 + \\ & \left[w_1^2 w_{11}^1 w_{12}^1 \left(\frac{tanh^{(2)}(b_1)}{2}\right) + w_2^2 w_{21}^1 w_{22}^1 \left(\frac{tanh^{(2)}(b_2)}{2}\right)\right] vgs.vds + \ldots \end{aligned} \tag{8}$$

Comparing (5) and (8) it can be seen that our developed neural network model is equivalent to the Volterra model or power-series expression commonly used for the current-voltage relationship in a FET. Moreover, the Volterra kernels of the series can now be easily calculated in function of the neural network parameters (the terms between brackets).

Equations (9) to (14) show how to calculate the kernels using the weights and bias values of the network. We only show the calculations for the impulse response, and the first and second order kemels, but the other formulas can be easily found. The corresponding first and second order derivatives have been developed.

$$Ids(Vgs0,Vds0) = h_0 = b_0 + \sum_{i=1}^{2} w_i^2 \tanh(b_i) \quad (9)$$

$$Gm1 = h_1(vgs) = \sum_{j=1}^{2} w_j^2 w_{j1}^1 (1 - \tanh^2(b_j)) \quad (10)$$

$$Gds = h_1(vds) = \sum_{j=1}^{2} w_j^2 w_{j2}^1 (1 - \tanh^2(b_j)) \quad (11)$$

$$Gm2 = h_2(vgs,vgs) = \frac{\sum_{j=1}^{2} w_j^2 w_{j1}^1 w_{j1}^1 (-2\tanh(b_j) + 2\tanh^3(b_j))}{2} \quad (12)$$

$$Gd2 = h_2(vds,vds) = \frac{\sum_{j=1}^{2} w_j^2 w_{j2}^1 w_{j2}^1 (-2\tanh(b_j) + 2\tanh^3(b_j))}{2} \quad (13)$$

$$Gmd = h_2(vgs,vds) = \frac{\sum_{j=1}^{2} w_j^2 w_{j1}^1 w_{j2}^1 (-2\tanh(b_j) + 2\tanh^3(b_j))}{2} \quad (14)$$

4. SIMULATION RESULTS

We have performed some simulations to show the validity of our approach. We have used the Curtice model to generate the training data, with 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 Levenverg-Marquardt algorithms, 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, using the formulas shown above. Then we have built the Volterra approximation (8) with the calculated kernels, and we have plotted it against the original FET behavior.

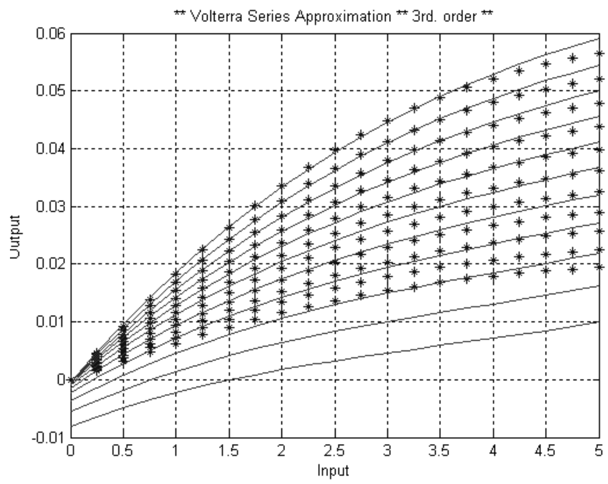

Figure 4. I/V Curves of the Curtice model (*) vs. I/V Curves of the Volterra-Neural Network based model (-), without normalized values

The results are reported in Figures 4 and 5 In Figure 4 we have trained the MLP model with the original input/output data, without normalizing the values. We can see that the Volterra approximation to the original data, built with the kernels calculated with the MLP model and including only up to the third order kernels, is quite accurate, but for the lower values of vgs there are some negative values which are not correct. However, in general, the form of the nonlinearity in this multivariable case is correct.

We have decided to try normalizing the input/output values to train the MLP model, to improve approximation, between the extremes of the domain interval of the hidden nodes activation function, the hyperbolic tangent. This way, the data are normalized before being used for the training, and de-normalized afterwards. The results have been impressively improved, and are shown in Figure 5, reaching a Mean Square Error (MSE) of $1x10^{-7}$.

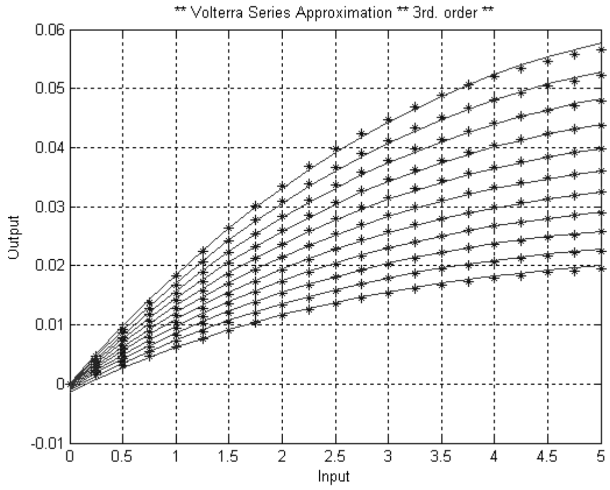

Figure 5. I/V Curves of the Curtice model (*) vs. I/V Curves of the Volterra-Neural Network based model (-) with normalized values

These results show the validity of our approach for the building of a multivariable Volterra series model of the nonlinearity in an electronic devices, and the results have shown also the importance of normalization for improving the Neural Networks training phase.

5. CONCLUSIONS

In this work, we have shown how a Neural Network model for a transistor nonlinearity and its analytical Volterra series model are totally equivalent. Therefore, we have explained here how it is possible to build the Volterra series analytical expression for an electronic device, even in the case of a multivariable nonlinearity, which is a difficult task, using parameters of a simple standard Neural Network model, trained only with voltage-current device measurements or simulation data.

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 quire accurate and fast. We have put in evidence also the importance of data normalization for the training phase on a neural network model, to improve accuracy in the final results obtained.

We want to highlight that our proposed approach implies and effective and concrete application of the neural networks models in the Electronics Field, that allows the building of an analytical Volterra series representation for a device, in this particular case a FET transistor, 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.

Our future work involves the implementation of this model inside a microwave circuit simulator.

References

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

[2] Boyd, S., Chua, L.O., Desder, C.A. Analytical Foundations of Volterra Series. *IMA Journal of Mathematical Control & Information*, No 1, (1984), 243-282.

[3] Curtice, W.R., Ettenberg, M. A Nonlinear GaAs FET Model for Use in the Design of Output Circuits for Power Amplifiers. *IEEE Transactions on Microwave Theory and Techniques*, Vol. 33, No. 12 (1985), 1383-1394.

[4] Garcia, J.A., Tazon Puente, A., Mediavilla Sanchez, A., Santamaria, I., Lazaro, M., Pantaleon, C.J., Pedro, J.C. 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), 261-276.

[5] Hakim, N.Z., Kaufman, J.J., Cerf, G., Meadows, H.E. Volterra characterization of Neural Networks. *Signals, Systems and Computers*, No. 2, (1991), 1128-1132.

[6] Harkouss, Y., Rousset, J., Chehade, H., Ngoya, E., Barataud, D., Teyssier, J.P. Modeling Microwave Devices and circuits for telecommunications systems design. *International Joint conference on Neural Networks*, No. 1, (1998), 128-133.

[7] Kashiwagi, H., Harada, H., Rong, L. 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), 707-712.

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

[9] Marmarelis, V.Z., Zhao, X. Volterra models and Three Layers Perceptron. *IEEE Transactions on Neural Networks*, Vol. 8, No. 6, (1997), 1421-1433.

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

[11] Soloway, D.I., Bialasiewicz, J.T. Neural Networks modeling of nonlinear systems based on Volterra series extension of a linear model. *IEEE International Symposium on Intelligent Control*, (1992), 7-12.

[12] Woods, Q.P., Batchelor, A.H., Johnson, S.R., Green, G.G.R. Comparison of the response of a time delay NN with an analytical model. *International Joint Conference on Neural Networks*, No. 2, (2002), 1120-1125,

[13] Wray, J., Green, G.G.R. Calculation of the Volterra kernels of non-linear dynamic systems using an artificial neural network. *Biological Cybernetics*, No. 71, (1994), 187-195.