Neural Networks for Large and Small-Signal Modeling of MESFET/HEMT Transistors: a Comparative Study

Marcelino Lázaro, Ignacio Santamaría, Carlos Pantaleón

DICOM, ETSII y Telecom, University of Cantabria
Avda. Los Castros, 39005, Santander, Spain

Abstract

In this paper we present a comparative study of three neural networks-based solutions for large and small-signal modeling of MESFET and HEMT transistors. The first two neural architectures are specific for this modeling problem: the Generalized Radial Basis Function (GRBF) network, and the Smoothed Piecewise Linear (SPWL) model. The results provided by these models are compared with those obtained by the well-known Multilayer Perceptron (MLP) network. Results are presented for both the large and small-signal regimes separately. Finally, a global model is proposed that is able to accurately characterize the whole behavior of the transistors. This model is based on a simple combination of the best models obtained for the two kinds of regimes.

1. Introduction

The design of microwave and millimeter-wave circuits and the increasing integration of hybrid and monolithic circuits has reinforced the need of accurate large and small-signal device models to improve the performance of these circuits and to minimize the number of design and fabrication steps required. Therefore, it is very important for efficient CAD tools to have good modeling approaches able to predict the small and large-signal nonlinear dynamic behavior of GaAs devices, such as metal semiconductor field effect transistor (MESFET), or high electron mobility transistor (HEMT).

Conventional nonlinear techniques applied to device modeling, such as closed-form equations [1,2], Volterra series [3], or the use of look-up tables [4], are difficult to implement in commercial simulators because of their high memory requirements or their computational burden.

Recently, some attempts have been made to model the nonlinear behavior of active devices and circuits by using neural networks [5,6]. Neural networks have the capability of approximating any nonlinear function and the ability to learn from experimental data; therefore, they are good candidates to solve device-modeling problems. However, practically all these neural approaches only consider the use of the Multilayer Perceptron (MLP) and, in this case, the memory requirements to give a good approximation, and the computational requirements to carry out the training of the network are high.

In order to avoid these problems we have proposed two models: the Generalized Radial Basis Functions (GRBF) network [7], for small-signal modeling, and the Smoothed Piecewise Linear (SPWL) model [8] for large-signal regimes. Both models require a low number of parameters and, at the same time, their computational requirements to train the models are lower than those of the above mentioned methods (including the MLP).

The paper is organized as follows. In Section 2 the problem of modeling microwave transistors is stated. In Section 3 the GRBF and SPWL models are described. In Section 4 the main results obtained are presented, and in Section 5 a global model is proposed, which characterize the whole device behavior. Finally, in Section 6, the main conclusions are exposed.

2. Modeling of MESFET and HEMT transistors

In this section we state the problems encountered when modeling microwave devices such as MESFET or HEMT transistors. Generally, to model a transistor there are two clearly different kinds of regimes, the large and the small-signal regimes, which usually are modeled separately.

2.1. Small-signal modeling of transistors

In a MESFET or a HEMT, the predominant nonlinear element is the drain-to-source current, I_{ds}, which depends on the drain-to-source, V_{ds}, and the gate-to-source, V_{gs}, bias voltages. This dependence is denoted as the *I/V* characteristic. As it is shown in [9], the *n*th-order

This work has been partially supported by the European Commission and CYCIT under grant 1FD97-1863-C02-01

intermodulation output power varies fundamentally as the square of the *n*th derivative of the *I/V* characteristic. Therefore, if we want to be able to model the small-signal intermodulation behavior, our model must accurately fit not only the nonlinear function but also its derivatives. In particular, when we apply a small-signal RF input around a bias point, the drain current I_{ds} depends on the bias point (V_{ds}, V_{gs}) and on the instantaneous small-signal voltages (v_{ds}, v_{gs}). We can approximate I_{ds} by the following truncated Taylor series expansion

$$\begin{aligned} I_{\mathrm{ds}} &= I_{\mathrm{dso}} + G_{\mathrm{m}} v_{\mathrm{gs}} + G_{\mathrm{ds}} v_{\mathrm{ds}} + G_{\mathrm{m2}} v_{\mathrm{gs}}^2 + G_{\mathrm{md}} v_{\mathrm{ds}} v_{\mathrm{gs}} \\ &\quad + G_{\mathrm{d2}} v_{\mathrm{ds}}^2 + G_{\mathrm{m3}} v_{\mathrm{gs}}^3 + G_{\mathrm{m2d}} v_{\mathrm{ds}} v_{\mathrm{gs}}^2 + G_{\mathrm{md2}} v_{\mathrm{ds}}^2 v_{\mathrm{gs}} + G_{\mathrm{d3}} v_{\mathrm{ds}}^3 \end{aligned} \qquad (1)$$

where I_{dso} is the dc drain current and (G_{m}, ... , G_{d3}) are coefficients related to the *n*th-order derivatives of the *I/V* characteristic evaluated at the bias point. Therefore, after obtaining a set of real measurements, our small-signal modeling problem consists in fitting a function (model) $\boldsymbol{G}_{SS} : \Re^2 \rightarrow \Re^{10}$, which approximates the nonlinear mapping from the input space of bias voltages, $\boldsymbol{V} = (V_{\mathrm{ds}}, V_{\mathrm{gs}})$, to the output space of model parameters $\boldsymbol{G}_{SS} = (I_{\mathrm{dso}}, G_{\mathrm{m}}, G_{\mathrm{ds}}, G_{\mathrm{m2}}, G_{\mathrm{md}}, G_{\mathrm{d2}}, G_{\mathrm{m3}}, G_{\mathrm{m2d}}, G_{\mathrm{md2}}, G_{\mathrm{d3}})$. Once this model is available, the drain current will be reconstructed by using the truncated Taylor series expansion (1).

2.2. Large-signal modeling of transistors

The large signal behavior of MESFET or HEMT transistors is governed by the nonlinear dynamic pulsed *I/V* characteristic that depends on the quiescent bias point [10]. Therefore, in this case the drain current I_{ds} depends on the bias point $(V_{\mathrm{ds}}, V_{\mathrm{gs}})$ and on the pulsed voltages $(v_{\mathrm{ds}}, v_{\mathrm{gs}})$ applied over the bias point. Now, our large-signal modeling problem consists in obtaining a function $\boldsymbol{G}_{LS} : \Re^4 \rightarrow \Re^1$, which approximates the nonlinear mapping from the input space of bias and pulsed voltages, $\boldsymbol{V} = (V_{\mathrm{ds}}, V_{\mathrm{gs}}, v_{\mathrm{ds}}, v_{\mathrm{gs}})$ to the output space $\boldsymbol{G}_{LS}(\boldsymbol{V}) = I_{\mathrm{ds}}$.

3. Proposed networks

In this section we describe the GRBF and SPWL networks that we have proposed to solve the above modeling problems.

3.1. Generalized Radial Basis Functions network

The Generalized Radial Basis Functions (GRBF) network is an extension of the Radial Basis Functions (RBF) network that relaxes the radial constraint for the basis kernels allowing different variances for each dimension of the input space. In that way it is possible to reduce the number of required basis functions, and therefore the number of parameters.To perform a general $\boldsymbol{G} : \Re^J \rightarrow \Re^M$ mapping, the k-th output of the GRBF network is given by

$$G_k(\boldsymbol{V}) = \sum_{i=1}^{I} g_i(\boldsymbol{V}) \qquad (2)$$

where *i* indexes the different GRBF units, $g_i(V) = \lambda_{ik}\, o_i(V)$ and $o_i(V)$ is the activation function of each unit

$$o_i(\boldsymbol{V}) = \prod_{j=1}^{J} \exp\text{-}\frac{\left(V_j - \mu_{i,j}\right)^2}{2\sigma_{i,j}^2} \qquad (3)$$

where V_j is the j-th element of input vector V.

The GRBF can be seen as a RBF network for which the Euclidean norm is replaced by a weighted norm. To show this, Eq (3) can be written as

$$o_i(\boldsymbol{V}) = exp\text{-}\left\|\boldsymbol{V} - \boldsymbol{\mu}_i\right\|_{W_i}^2 \qquad (4)$$

where

$$\left\|\boldsymbol{V} - \boldsymbol{\mu}_i\right\|_{W_i}^2 = (\boldsymbol{V} - \boldsymbol{\mu}_i)^{\mathrm{T}} \boldsymbol{W}_i^{\mathrm{T}} \boldsymbol{W}_i\ (\boldsymbol{V} - \boldsymbol{\mu}_i) \qquad (5)$$

and $\boldsymbol{W}_i$ is a diagonal matrix given by

$$\boldsymbol{W}_i = \frac{1}{\sqrt{2}}\, diag\left(\frac{1}{\sigma_{i1}}, \ldots, \frac{1}{\sigma_{iJ}}\right) \qquad (6)$$

Restricting this matrix to be diagonal, we are allowing different variances along each input dimension, but we do not allow the elliptic basis functions to rotate.

The basis functions are initialized by a variant of the Orthogonal Least Squares (OLS) algorithm [11] which is able to work with elliptical kernels. The error function to be minimized is the quadratic error

$$E = \sum_p \sum_k \left(y_k(\boldsymbol{V}_p) - G_k(\boldsymbol{V}_p)\right)^2 \qquad (7)$$

and the variances and the centers of the network are adapted by using the following equations of the gradient

$$\frac{\partial E}{\partial \sigma_{ij}} = -2 \sum_p \sum_k e_k(\boldsymbol{V}_p)\, o_i(\boldsymbol{V}_p)\, \lambda_{ik}\, \frac{1}{\sigma_{ij}} \left(\frac{V_{pj} - \mu_{ij}}{\sigma_{ij}}\right)^2 \qquad (8)$$

$$\frac{\partial E}{\partial \mu_{ij}} = -2 \sum_p \sum_k e_k(\boldsymbol{V}_p)\, o_i(\boldsymbol{V}_p)\, \lambda_{ik}\, \frac{1}{\sigma_{ij}} \left(\frac{V_{pj} - \mu_{ij}}{\sigma_{ij}}\right) \qquad (9)$$

where *p* indexes the input patterns, *k* the output dimensions, and $y_k(\boldsymbol{V}_p)$ and $e_k(\boldsymbol{V}_p)$ are the desired output and the network error respectively, of the *k*-th output dimension for the *p*-th input pattern. With the centers and variances fixed, the dependence with the λ_{ik} parameters is linear, and therefore its optimum values are easily calculated by least-squares. This adaptation process is iteratively repeated until a suitable error is reached.

3.2. Smoothed Piecewise Linear model

The Smoothed Piecewise Linear (SPWL) model is an extension of the well-known Canonical Piecewise Linear model proposed by Chua [12]. This model, as it is shown in [13], can be seen as a neural network. Basically, the model implements a general mapping $\boldsymbol{G} : \Re^M \to \Re^N$ as follows

$$\boldsymbol{G}(\boldsymbol{V}) = \boldsymbol{a} + \boldsymbol{B}\boldsymbol{V} + \sum_{i=1}^{\theta} \boldsymbol{c}_i \left| \left\langle \boldsymbol{\alpha}_i , \boldsymbol{V} \right\rangle - \beta_i \right| \tag{10}$$

where $\boldsymbol{V}$ and α_i are vectors of dimension M, $\boldsymbol{a}$ and $\boldsymbol{c}_i$ are vectors of dimension N, $\boldsymbol{B}$ is an $N \times M$ matrix, β_i is scalar and < , > denotes the inner product. The model divides the input space into different regions by means of several boundaries implemented by hyperplanes of dimension $M-1$ (defined by the expression inside the absolute value function), and it carries out the function approximation by means of the combination of hinging hyperplanes of dimension M, which are the result of joining linear hyperplanes over the boundaries defined in the input space. In any region the model is composed by a linear combination of linear hyperplanes, and the transitions at the boundaries are governed by the absolute value function. Therefore, this model inherits some properties from the absolute value function: it is continuous but not derivable along the boundaries. Moreover, the second and higher order derivatives are zero except at the boundaries where they are discontinuous.

To overcome this drawback, the SPWL model substitutes the absolute value function for a smooth and derivable function in order to smooth the transition at the boundaries dividing the input space. Several possibilities exist to smooth the absolute value function allowing, at the same time, a parametric control of the "sharpness" of the transition. We have proposed the following smoothing function

$$lch(x,\gamma) = \frac{1}{\gamma} \ln\left(\cosh(\gamma\, x) \right) \tag{11}$$

where γ is a parameter that allows to control the smoothness of the transition. There are several reasons to select this function. For instance, its derivatives do not present overshootings unlike some other commonly used smoothing functions ($x$tanh(x), for instance): this is a clear advantage when we try to fit both a function and its derivatives. In the other hand, the first derivative of (11) is $lch'(x,\gamma) = \tanh(\gamma\, x)$, which is the activation function of a universal approximator such as the MLP. Finally, the proposed SPWL model is given by

$$\boldsymbol{G}(\boldsymbol{V}) = \boldsymbol{a} + \boldsymbol{B}\boldsymbol{V} + \sum_{i=1}^{\theta} \boldsymbol{c}_i lch(\left\langle \alpha_i , \boldsymbol{V} \right\rangle - \beta_i , \gamma_i) \tag{12}$$

The training of this network is carried out by means of an optimization method equivalent to the proposed by Chua for the Canonical Piecewise Linear model [12].

Let us consider that we want to approximate a mapping $\Re^M \to \Re$ using a set of N input-output samples $(\boldsymbol{V}_l, y_l)$, l=1,...,N with $\boldsymbol{V}_l=(V_{1,l}, V_{2,l},..., V_{\mathrm{M},l})$. Assuming that $\alpha_{\imath,M} \neq 0$, we can eliminate one coefficient from each boundary by rewriting $<\alpha_i, \boldsymbol{V}> - \beta_i$, as

$$b_i(\boldsymbol{V}) = m_{i,1}V_1 + m_{i,2}V_2 + \cdots + m_{i,M-1}V_{M-1} - V_M + t_i \tag{13}$$

where $b_i(\boldsymbol{V})$ denotes the ith boundary evaluated at $\boldsymbol{V}$. Finally, taking into account that $\boldsymbol{B}$ and $\boldsymbol{a}$ in (7) are now a vector $\boldsymbol{b}=(b_1,...,b_M)^T$, and a scalar a, respectively; our generic SPWL model, with θ boundaries, can be written as

$$G(\boldsymbol{V}) = a + \boldsymbol{b}^T\boldsymbol{V} + \sum_{i=1}^{\theta} c_i \; lch(b_i(\boldsymbol{V}) \, , \gamma) \tag{14}$$

The model parameters can be grouped into two vectors: $\boldsymbol{z_p}$ grouping the coefficients associated to the linear combination of components,

$$\boldsymbol{z_p} = \left(a, b_1, \ldots, b_M, c_1, \ldots c_\theta\right)^T \tag{15}$$

and $\boldsymbol{z_r}$, grouping the parameters defining the boundaries of the domain space

$$\boldsymbol{z_r} = \left(m_{1,1}, \ldots, m_{1,M-1}, \ldots, m_{\theta,1}, \ldots m_{\theta,M-1}, t_1, \ldots t_\theta\right)^T \tag{16}$$

The error function to be minimized is given by

$$E(\boldsymbol{z}_p, \boldsymbol{z}_r) = \sum_{l=1}^{N} \left(y_l - \left(a + \boldsymbol{b}^T\boldsymbol{V}_l + \sum_{i=1}^{\theta} c_i \; lch(b_i(\boldsymbol{V}_l), \gamma) \right) \right)^2 \tag{17}$$

The algorithm begins by fixing the initial location of each partition boundary, i.e., the vector $\boldsymbol{z_r}$. Generally, they are chosen randomly. Then, the approximation error $E(\boldsymbol{z_p}, \boldsymbol{z_r})$ is a quadratic function of $\boldsymbol{z_p}$, and its minimum is given by

$$z_p = \left(\boldsymbol{A}\boldsymbol{A}^T\right)^{-1} \boldsymbol{A}\boldsymbol{y} \tag{18}$$

where $\boldsymbol{y} = (y_1, \ldots, y_N)^T$, $\boldsymbol{A}$ is the following M+θ+1×N matrix, which can be partitioned as

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{1} \\ \boldsymbol{U} \\ \boldsymbol{W} \end{bmatrix} \tag{19}$$

where **1** denotes a row of N ones, $\boldsymbol{U}$ is an M×N matrix with elements, $u_{i,j} = x_{i,j}$, and $\boldsymbol{W}$ is an θ×N matrix with elements $w_{i,j} = lch(b_i(\boldsymbol{V}_j), \gamma)$.

Once the optimal z_p parameters (for a given initial z_r partition) are calculated, the algorithm estimates a new optimal partition z_r. This partition is found by calculating the gradient $\boldsymbol{g}$ and the Hessian $\boldsymbol{Y}$, which specify the optimal searching direction to modify z_r according to

$$\boldsymbol{s} = -\boldsymbol{Y}^{-1}\boldsymbol{g} \tag{20}$$

The gradient, $\boldsymbol{g}$ and the Hessian $\boldsymbol{Y}$ are given by

$$\boldsymbol{g} = 2\boldsymbol{KGe}\,, \tag{21}$$

$$\boldsymbol{Y} = 2\boldsymbol{KGG}^T\boldsymbol{K} + 2\boldsymbol{K}\frac{\partial\boldsymbol{G}}{\partial z_r}\boldsymbol{e}\,, \tag{22}$$

where $\boldsymbol{e} = (e_1, \ldots, e_N)^T$ is the vector of errors, $\boldsymbol{K}$ is given by

$$\boldsymbol{K} = diag(\underbrace{c_1,\ldots,c_1}_{M\text{-1 terms}}\underbrace{c_2,\ldots,c_2}_{M\text{-1 terms}}\ldots\underbrace{c_\theta,\ldots,c_\theta}_{M\text{-1 terms}}, c_1, c_2, \ldots, c_\theta) \tag{23}$$

and $\boldsymbol{G}$ is the following matrix

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G}^1 \\ \vdots \\ \boldsymbol{G}^\theta \\ \boldsymbol{P} \end{bmatrix} \tag{24}$$

where $\boldsymbol{G}^{\mathrm{k}}$ are $M\text{-}1 \times N$ matrices with elements $g^k_{i,j} = V_{i,j}\tanh(\gamma b_k(\boldsymbol{V}_j))$, and $\boldsymbol{P}$ is an $\theta \times N$ matrix with elements $p_{i,j} = \tanh(\gamma b_i(\boldsymbol{V}_j))$.

The second term of (22) involves the second derivative of the SPWL model, $\mathrm{sech}^2(b_i(\boldsymbol{V}_l))$, which is a localized function along the boundaries: only points close to the boundaries contribute to this term. In practice, it has been observed that a great computational saving (without any noticeable degradation) can be achieved by dropping out this term from the Hessian, that is, we use $\boldsymbol{Y} = 2\boldsymbol{KGG}^T\boldsymbol{K}$. Once the search direction (20) has been calculated, the new boundaries are estimated as

$$z = z + \alpha\boldsymbol{s} \tag{25}$$

where α = argmin(E(z_p,z_r+αs)). With this new partition the process is repeated: the optimal coefficients z_p are calculated for these new boundaries, and then the optimal partition is reestimated again, until a given error is reached.

4. Results

In this section we present the results obtained with the two networks described in the previous section, and they are compared with those provided by the Multilayer Perceptron (MLP) network as an appropriate reference. The three models have been used to model a NE72084 MESFET from experimental measurements. Moreover, a brief study of the computational burden needed for the models is also presented.

4.1. Small-Signal Results

For the small-signal model we dispose of a set of measurements of the parameters included in the Taylor series (3), $(I_{\mathrm{dso}}, G_{\mathrm{m}}, G_{\mathrm{ds}}, G_{\mathrm{m2}}, G_{\mathrm{md}}, G_{\mathrm{d2}}, G_{\mathrm{m3}}, G_{\mathrm{m2d}}, G_{\mathrm{md2}}, G_{\mathrm{d3}})$, which compose the output of the small-signal model. In a first approach we use an individual network to model each output of this model. This approach leads to the results presented in Table I. The accuracy of the models is measured in terms of the signal to noise (SNR) ratio, in dB, for each scalar output. It can be seen that the SPWL and the GRBF model provide better results than the MLP in almost all the functions to be modeled. However, there is not a network that provides the best results globally. This is due to the different nature of the basis functions of each network. Depending on the shape of the function to be modeled, a different basis function is more suitable. Therefore, and obvious solution would be to use a mixed model combining these two kind of networks, selecting for each output the network providing the best results. This solution, although provides good results, presents a relatively high number of parameters to be implemented in a simulator. In order to reduce the number of parameters, a second approach consists in using a common network to perform the whole mapping globally. Table II presents the results obtained with such a solution. It can be seen that, in this case, the SPWL model provides globally the best results. These results are slightly lower than those provided by the previous solution, but the saving in the number of parameters is relevant. In general terms, the second approach seems to be more reasonable because of the economy of requirements to its implementation in commercial simulators, but for some specific applications that could require a very high accuracy, the first approach can be employed.

4.2. Large-Signal Results

For the large-signal model, the results obtained using the different neural models are shown in Table III. The results presented correspond to solutions with a low number of parameters to facilitate the implementation of the solution in simulators. In this case, the SPWL model clearly provides the best results. The GRBF network, on the other hand, provides results similar to the MLP.

	N_{param}	I_{ds}	G_{d2}	G_{d3}	G_{ds}	G_{m1}	G_{m2}	G_{m2d}	G_{m3}	G_{md}	G_{md2}
MLP(6)	25x10	38.4	24.9	20.1	37.6	42.4	38.4	21.5	21.2	32.4	20.1
GRBF(5)	25x10	44.3	29.1	21.5	38.8	43.7	34.3	22.2	25.2	34.3	22.2
SPWL(6)	22x10	45.4	28.8	19.8	43.3	46.5	36.7	22.7	23.0	35.2	19.9

Table I: Small-signal results using individual networks for each output. The number between parentheses indicates the number of basis functions employed for each model

	N_{par}	I_{ds}	G_{d2}	G_{d3}	G_{ds}	G_{m1}	G_{m2}	G_{m2d}	G_{m3}	G_{md}	G_{md2}
MLP(8)	114	29.8	17.9	17.7	34.0	36.2	29.0	20.2	19.6	24.6	17.6
GRBF(8)	114	29.7	17.5	18.5	30.4	31.0	26.2	22.4	22.0	24.2	18.9
SPWL(7)	115	36.1	17.6	17.6	33.6	39.6	27.1	20.4	17.8	29.1	18.5
MLP(11)	153	30.9	22.8	19.2	34.6	37.9	29.0	23.2	21.4	25.8	18.4
GRBF(11)	154	32.2	19.4	19.7	33.7	36.0	31.2	23.5	23.0	29.1	21.5
SPWL(10)	151	42.3	27.6	18.8	41.2	41.6	31.7	22.3	21.0	32.1	22.0

Table II: Small-signal results for a single network.

Model	Parameters	SNR
MLP (5)	31	24.60 dB
GRBF (4)	36	24.95 dB
SPWL (5)	31	28.70 dB
MLP (9)	55	26.44 dB
GRBF (6)	54	27.55 dB
SPWL (10)	56	31.58 dB
MLP (11)	67	27.36 dB
GRBF (7)	63	28.00 dB
SPWL (12)	66	32.67 dB

Table III: Large-Signal results

4.3. Computational Burden of the Models

Another interesting aspect to be considered is the computational burden associated to each model. Here we present a brief study of this cost presenting the training time of each model. The simulations have been obtained working with MATLAB® in a Pentium Celeron PC. The SPWL model obtains the fastest training. To give an idea of the training time differences, for example, a mean value of approximately 14 minutes is required for the SPWL(12) to converge in the large signal model in order to obtain the results presented in Table III. In the same training conditions, the GRBF(7) requires roughly 32 minutes and the MLP(11) needs more than 1 hour and a half. In the small signal modeling with a single network, the SPWL(7) required a mean of 7 minutes, while the GRBF(8) needed roughly 15 minutes, and the MLP(8) 47 minutes approximately. Therefore, with respect to the computational burden, the SPWL network presents the fastest training, and the GRBF is faster than the MLP network. Moreover, it can be taken into account that all these networks present the problem of local minima, and several simulation have to be performed to obtain a suitable solution, which increases the penalization for a slow training.

5. Global Model Proposed

In the previous sections we have treated the problem of modeling a transistor as it is usually faced: the large and small-signal behaviors of the transistors are modeled separately. We have obtained a model for each regime. Therefore, one model must be selected in function of the kind of regime in which the transistor will work.

In order to avoid this, we propose to use a single global model to completely characterize the transistor behavior. This model consists in a combination of the two best sub-models previously obtained: one for the large-signal regime, and other one for the small-signal regime. The global model is performed weighting the outputs of the two submodels in a reasonable way. The first and simplest approach would be to define an appropriate boundary between the large and the small signal regimes and make a “hard” decision between the two sub-models. A more elaborate alternative would be to implement a smooth transition between the two regimes. In any case, this kind of solution provides a single model that would be able to adequately characterize the whole behavior of a transistor. Fig. 1 shows a block diagram of this global model.

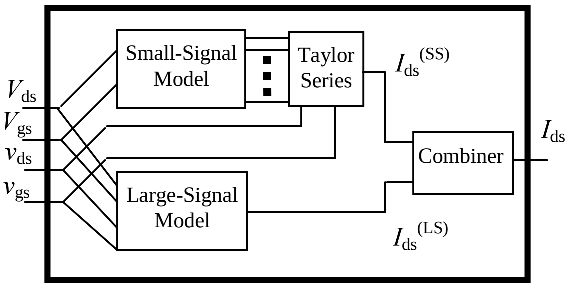

Figure 1: Global Model. The model combines the outputs of the small and large-signal submodels

6. Conclusions

In this paper we have presented a comparative study of several neural networks solutions for the large and small signal modeling of MESFET and HEMT transistors.

For the large-signal behavior, the SPWL model provides clearly the best results. For the small-signal regime, when using a single network, the SPWL model provides also the best solution globally. But when a very high accuracy is needed, the option of a mixed model, using independent SPWL and the GRBF networks for each output, can be employed, with the price of a higher number of parameters.

Relative to the computational burden, the SPWL has shown to be the more economic network, and the GRBF network, in any case, requires a lower computational burden than the MLP.

Finally, we have proposed a global model combining in a simple way the sub-models obtained for both the large and small-signal regimes. This model allows a whole characterization of the device, avoiding the need of working with several models for the same device.

References

[1] W.R. Curtice, M. Ettemberg, "A nonlinear GaAs FET model for use in the design of output circuits for power amplifiers", IEEE Trans. MTT., vol. 33, no. 6, pp. 1383-1394, 1985.

[2] A. McCamant, G. McCormak, D. Smith, "An improved GaAs MESFET for SPICE", IEEE Trans Microwave Theory Technol., vol. 38, no. 6, pp. 822-824, 1990.

[3] S. A. Maas, A. Crosmun, "Modeling the gate I/V characteristic of a GaAs MESFET for Volterra-series analysis", IEEE Trans. Microwave Theory Technol., vol. 37, no. 7, pp. 1134-1136, 1989.

[4] D. Root, S. Fan, J. Meyer, "Technology independent large signal nonquasi-static FET models by direct construction from automatically characterized device data", Proc. 21st European Microwave Conf., Stuttgart, Germany, pp. 923-927, 1991.

[5] J. Rousset, et al. "An accurate neural network model of FET for intermodulation and power analysis", Proc. 26th European Microwave Conf., Prague, Czechoslovakia, 1996.

[6] K.Shirakawa, et al. "A large signal characterization of a HEMT using a multilayered neural network", IEEE Trans. Microwave Theory Technol., vol. 45, no. 9, pp. 1630-1633, 1997.

[7] I. Santamaría et al. " A nonlinear MESFET model for intermodulation analysis using a generalized radial basis function network", Neurocomputing, vol. 25, pp. 1-18, 1999.

[8] M. Lázaro, I. Santamaría, C. Pantaleón, "Smoothing the Canonical Piecewise Linear Model: An efficient and derivable large-signal model for MESFET/HEMT transistors". Submitted to IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications.

[9] A. M. Crosmun, S. Maas, "Minimization of intermodulation distortion in GaAs MESFET small-signal amplifiers", IEEE Trans. Microwave Theory Technol., vol. 37, no. 9, pp. 1411-1417, 1989.

[10] T. Fernández et al. "Extracting a Bias-Dependent Large Signal MESFET Model from I/V measurements". IEEE Trans. on MTT , vol 34, no 3, pp 372-378, March 1986.

[11] S. Chen, C.F. Cowan, P.M. Grant, "Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks", IEEE Trans. on Neural Networks, vol 2, no 2, pp 302-309, March 1991.

[12] L. O. Chua, A. C. Deng, "Canonical piecewise-linear modeling", IEEE Trans. Circuits Syst., vol. 33, no. 5, pp. 511-525, 1986.

[13] R. Batruni, "A multilayer neural network with piecewise-linear structure and back-propagation learning", IEEE Trans. on Neural Networks, vol. 2, pp. 395-403, 1991.