A SMOOTH AND DERIVABLE LARGE-SIGNAL MODEL FOR MICROWAVE HEMT TRANSISTORS

M. Lázaro, I. Santamaría, C. Pantaleón

Dept. Ingeniería de Comunicaciones, Universidad de Cantabria
Av. Los Castros s/n, 39005, Santander, SPAIN

ABSTRACT

In this paper we present the Smoothed Piecewise Linear (SPWL) model as a useful tool for device modeling problems. The SPWL model is an extension of the well-known canonical piecewise linear model proposed by Chua, which substitutes the abrupt absolute value function for a smoothing function (the logarithm of hyperbolic cosine). This function makes the model derivable; moreover the smoothness of the global model can be controlled by means of a single smoothing parameter. The parameters of the model are adapted to fit the nonlinear function, while the smoothing parameter is selected according to derivative constraints. The proposed SPWL model is successfully applied to model a microwave HEMT transistor under optical illumination using real measurements.

1. INTRODUCTION

In recent years, the need for efficient CAD tools to facilitate the design and fabrication of microwave circuits has spurred the search for new nonlinear models to predict the small and large-signal nonlinear dynamic behavior of microwave and milimeter-wave active devices such as MESFET or HEMT transistors [1].

Conventional nonlinear techniques for device modeling, such as closed-form equations [2], Volterra series [3], or look-up tables [4], suffer the drawback of a high memory requirement or a high computational burden and, therefore, their implementation in commercial simulators is difficult. Moreover, these techniques do not represent adequately the nonlinear function derivatives around the bias point, which is interesting, for instance, to model the intermodulation distortion behavior. Although this is not essential when modeling transistors in a large-signal regime, to get a smooth and derivable model leads to a better representation of the device behavior.

Recently, some alternatives based on neural networks have been proposed for this problem. Neural networks have the capability of approximating any nonlinear function and they learn from experimental data; therefore, they are specially suited for this modeling problem. However, most of these neural approaches are based on the Multilayer Perceptron (MLP) [5], which also requires a large number of parameters for accurate modeling and it has the drawback of slow learning. Other architectures have been proposed for specific problems. For instance, in [6] the Generalized Radial Basis Function (GRBF) network was proposed to model the derivatives around the bias point for small-signal intermodulation prediction. Nevertheless, for large-signal modeling problems, the semilocal activation function of the GRBF network increases the number of parameters .

An alternative method widely used in device modeling is the Canonical Piecewise Linear (PWL) model proposed by Chua [7]. It provides accurate approximations with a low number of parameters and with a computational burden lower than the neural networks solutions. The drawback of this model is that it lacks the capability of approximating the derivatives of the function because of its piecewise nature: the second and higher order derivatives are always zero.

In this paper we propose the Smoothed Piecewise Linear (SPWL) model as a generalization of the PWL model, which, retaining its advantages, makes it smooth and derivable. In the SPWL model, the absolute value function used in the PWL, is replaced by a smooth function (the logarithm of hyperbolic cosine). A single parameter controls the smoothness (i.e., the integral of the squared second derivative) of the whole model: this fact provides a link with regularization theory. The proposed SPWL is applied to approximate the nonlinear I/V characteristic of a microwave HEMT.

2. THE SMOOTHED PWL MODEL

2.1 Smoothing the canonical PWL model

In its basic formulation, the canonical PWL model, proposed by Chua [7], performs a mapping $\boldsymbol{f} : \boldsymbol{R}^M \rightarrow \boldsymbol{R}^N$ as follows

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

where $\boldsymbol{x}$ 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, and it carries out the function approximation by means of the combination of hinging hyperplanes of dimension M. Such hinging hyperplanes are the result of joining linear hyperplanes over the boundaries defined in the input space. In (1), it can be seen that the expression inside

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

the absolute value function defines the boundaries partitioning the domain space. The canonical PWL 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, in this paper we substitute the absolute value function for a derivable function in order to smooth the joint of hyperplanes at the boundaries. Several possibilities exists to smooth the absolute value function allowing, at the same time, a parametric control of the "sharpness" of the transition. In this paper we propose the following smoothing function

$$lch(x,\gamma) = \frac{1}{\gamma}\ln(\cosh(\gamma x)) \qquad (2)$$

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 (2) 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

$$f(\boldsymbol{x}) = \boldsymbol{a} + \boldsymbol{Bx} + \sum_{i=1}^{\theta} c_i lch(\langle \alpha_i, \boldsymbol{x}\rangle - \beta_i, \gamma_i) \ . \qquad (3)$$

2.2 Smoothness of the SPWL model

Standard regularization techniques minimize a cost functional consisting of two terms: the first one measures the closeness to the data, and the second term weights the cost associated with a functional that measures the smoothness of the solution, i.e.,

$$E = \sum_{\mathrm{i}} (f(x_i) - y_i)^2 + \lambda \|Pf(x)\|^2 \qquad (4)$$

where y_i are the measurements, λ is a regularization parameter, which controls the compromise between degree of smoothness of the solution and its closeness to the data, and P is a functional (stabilizer). Smoothness can be measured in a number of different ways, generally, the stabilizer P involves some derivatives of the function. A widely used class of stabilizers is given by the following functionals [8]

$$\|P^m f\|^2 = \sum_{\mathrm{i}_1 \ldots \mathrm{i}_m}^{\mathrm{n}} \int_{R^{\mathrm{n}}} \mathrm{d}x \, (\partial_{\mathrm{i}_1 \ldots \mathrm{i}_m} f(x))^2 \qquad (5)$$

where $\partial_{i_1 \ldots i_m} = \partial^m / \partial x_{i_1} \ldots \partial x_{i_m}$ and $\mathrm{m} \geq 1$.

For the SPWL model, the second derivative of each unit is a localized function, with the shape $\gamma\,\mathrm{sech}^2(\gamma x)$. Therefore, we can assume that the second derivative of each unit only overlaps with the nearest component. Without lack of generality, we can assume that the model is composed of two weighted components, separated a distance b. Let us consider a 1-dimensional SPWL model using the same γ for all the boundaries (breakpoints), then

$$f(x) = c_1 lch(x,\gamma) + c_2 lch(x-b,\gamma) \ . \qquad (6)$$

For this 1-dimensional case, the stabilizer has the particular expression of the squared second derivative of the model, which is given by

$$\|P^1 f\|^2 = \frac{4}{3}(c_1^2 + c_2^2)\gamma + 8c_1c_2\,\gamma\,\mathrm{cosech}^2(\gamma b)\left[\frac{\gamma b}{\tanh(\gamma b)} - 1\right] \qquad (7)$$

which varies monotonically with γ, independently of b and the weighting parameters c_1 and c_2. Equation (7) formalizes somehow the expected behavior of the SPWL model: a small γ forces very smooth transitions and therefore increases the smoothness of the global model. The same behavior can be obtained for an input space of higher dimensionality.

The above result establishes a link between the SPWL model and regularization theory. Specifically, the smoothing parameter γ can be seen as a standard regularization parameter. In this way, to minimize the squared error with a proper selection of γ, according to some smoothness constraint (for instance derivative constraints), is equivalent to minimize a regularized functional such as (4).

3. MODEL TRAINING

3.1 Boundaries and hyperplanes estimation

The method applied for estimating the boundaries and hyperplanes is equivalent to the optimization method proposed by Chua for the canonical PWL model [7]. Here we reformulate this method for a SPWL model. Let us consider that we want to approximate a mapping $R^M \to R$ using a set of N input-output samples $(\boldsymbol{x}_l, y_l)$, $l=1,\ldots,N$ with $\boldsymbol{x}_l=(x_{1,l}, x_{2,l},\ldots, x_{\mathrm{M},l})$. Assuming that $\alpha_{t,M} \neq 0$, we can eliminate one coefficient from each boundary by rewriting $<\alpha_i, \boldsymbol{x}> - \beta_i$, as

$$b_i(\boldsymbol{x}) = m_{i,1}x_1 + m_{i,2}x_2 + \cdots + m_{i,M-1}x_{M-1} - x_M + t_i \qquad (8)$$

where $b_i(\boldsymbol{x})$ denotes the ith boundary evaluated at $\boldsymbol{x}$. Finally, taking into account that $\boldsymbol{B}$ and $\boldsymbol{a}$ in (3) are now a vector $\boldsymbol{b}=(b_1,\ldots,b_M)^T$, and a scalar a, respectively; our generic SPWL model, with θ boundaries, can be written as

$$f(\boldsymbol{x}) = a + \boldsymbol{b}^T \boldsymbol{x} + \sum_{i=1}^{\theta} c_i \, lch(b_i(\boldsymbol{x}), \gamma) \ . \qquad (9)$$

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

$$\boldsymbol{z_p} = (a, b_1, \ldots, b_M, c_1, \ldots c_\theta)^T \qquad (10)$$

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

$$\boldsymbol{z_r} = (m_{1,1}, \ldots, m_{1,M-1}, \ldots, m_{\theta,1}, \ldots m_{\theta,M-1}, t_1, \ldots t_\theta)^T \qquad (11)$$

The error function to be minimized is given by

$$E(z_p, z_r) = \sum_{l=1}^{N}\left(y_l - \left(a + \boldsymbol{b}^T \boldsymbol{x}_l + \sum_{i=1}^{\theta} c_i \; lch\big(b_i(\boldsymbol{x}_l), \gamma\big) \right)\right)^2 \quad (12)$$

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 $E(z, z_r)$ is a quadratic function of z, and its minimum is given by

$$z_p = \left(\boldsymbol{A}\boldsymbol{A}^T\right)^{-1}\boldsymbol{A}\boldsymbol{y} \quad (13)$$

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

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{1} \\ \boldsymbol{U} \\ \boldsymbol{V} \end{bmatrix} \quad (14)$$

where $\mathbf{1}$ denotes a row of N ones, $\boldsymbol{U}$ is an $M\times N$ matrix with elements, $u_{i,j} = x_{i,j}$, and $\boldsymbol{V}$ is an $\theta\times N$ matrix with elements $v_{i,j} = lch\big(b_i(\boldsymbol{x}_j), \gamma\big)$.

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

$$\boldsymbol{s} = -\boldsymbol{Y}^{-1}\boldsymbol{g} \; . \quad (15)$$

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

$$\boldsymbol{g} = 2\boldsymbol{K}\boldsymbol{G}\boldsymbol{e}\,, \quad (16)$$

$$\boldsymbol{Y} = 2\boldsymbol{K}\boldsymbol{G}\boldsymbol{G}^T\boldsymbol{K} + 2\boldsymbol{K}\frac{\partial \boldsymbol{G}}{\partial z_r}\boldsymbol{e}\,, \quad (17)$$

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

$$\boldsymbol{K} = diag(\underbrace{c_1,\ldots,c_1}_{M\text{-1 terms}}\underbrace{c_2,\ldots,c_2}_{M\text{-1 terms}}\ldots\underbrace{c_\theta,\ldots,c_\theta}_{M\text{-1 terms}}, c_1, c_2, \ldots, c_\theta) \quad (18)$$

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

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G}^1 \\ \vdots \\ \boldsymbol{G}^\theta \\ \boldsymbol{P} \end{bmatrix} \quad (19)$$

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

The second term of (17) involves the second derivative of the SPWL model, $\operatorname{sech}^2(b_i(x_l))$, 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 this term from the Hessian, that is, we use $\boldsymbol{Y} = 2\boldsymbol{K}\boldsymbol{G}\boldsymbol{G}^T\boldsymbol{K}$. Once the search direction (15) has been calculated, the new boundaries are estimated as

$$z_r = z_r + \alpha \boldsymbol{s} \quad (20)$$

where $\alpha = \operatorname{argmin}(E(z_p, z_r + \alpha s))$. With this new partition the process is repeated: the optimal coefficients $\boldsymbol{z_p}$ are calculated for these new boundaries, and then the optimal partition is reestimated again, until a given error is reached.

3.2 Estimation of the smoothing parameter

For the estimation of γ, which is a key parameter of the SPWL model, several strategies are possible. The simplest one is to minimize (12) applying a gradient descent algorithm. A different γ can be used for each boundary; however, the improvement achieved over using a common γ for all the boundaries does not compensate for the increase in the number of model parameters.

A more interesting alternative for device modeling problems is to use information about the function derivatives. As it was said before, to reproduce the intermodulation distortion behavior it is necessary to model the higher order derivatives. It seems reasonable, therefore, to look for a tradeoff between the approximation of the function and the approximation of these derivatives. For example, let us assume that it is possible to measure the first derivative of the model with respect to the jth input parameter: $\boldsymbol{y}' = (y_1, \ldots, y_N)^T$, then γ can be selected to minimize the error of the model over this derivative. In this case, the optimal γ is obtained by applying

$$\gamma_{k+1} = \gamma_k + \mu \sum_{l=1}^{N} e'_l \left(\sum_{i=1}^{\theta} c_i m_{i,j} b_i(\boldsymbol{x}_l) \operatorname{sech}^2\big(\gamma b_i(\boldsymbol{x}_l)\big) \right) \quad (21)$$

As a conclusion of this section, we can say that one of the most relevant characteristics of the SPWL model is that we can take advantage of the additional degree of freedom provided by γ to fit the derivatives without degrading noticeably the fit to the function

4. SPWL LARGE-SIGNAL HEMT MODEL

In this section we use the proposed SPWL model to characterize the large-signal behavior of a microwave HEMT Philips D02AH (4*30μm) transistor under optical illumination. The SPWL model is used to accurately reproduce the nonlinear dependence of the drain to source current I_{ds} with respect to the bias voltages (V_{gs}, V_{ds}), the instantaneous voltages (v_{gs}, v_{ds}) and the incident optical power P_o. We dispose of a set of 9792 input/output samples, which have been used to train and test the SPWL model and four additional models used for comparison: a MLP network, a GRBF network, the canonical piecewise linear (CPWL) and an analytical model [9].

Table I presents the signal to noise ratio (SNR) values in dB for the I_{ds} estimates and for the first derivatives with respect to the instantaneous voltages (v_{gs}, v_{ds}). From the Table I we can see that for a given number of parameters, the SPWL model provides the best results for the function and specially for the derivatives.

Figure 1 shows the original function and derivatives (left) and the approximation given by the SPWL model (right) for an incident optical power $P_o = 1\ mW$, and bias voltages $V_{gs} = 0V$ and $V_{ds} = 2\ V$.

In this section, we have obtained large-signal SPWL models with a small number of parameters (from 5 to 12 hyperplanes). In this case, the model is able to fit only the first derivative (see Fig. 1), since the higher derivatives are very spiky. In order to improve the behavior of the second and third derivatives it is necessary to increase the number of parameters (hyperplanes) of the model.

Another important advantage of the SPWL model is its low computational burden. Using MATLAB in a Pentium 200MHz, to train a SPWL model requires a few minutes, while the MLP or the GRBF need several hours, and the analytical model requires a exhaustive study of the problem that, moreover, is device dependent and must be repeated for each new device.

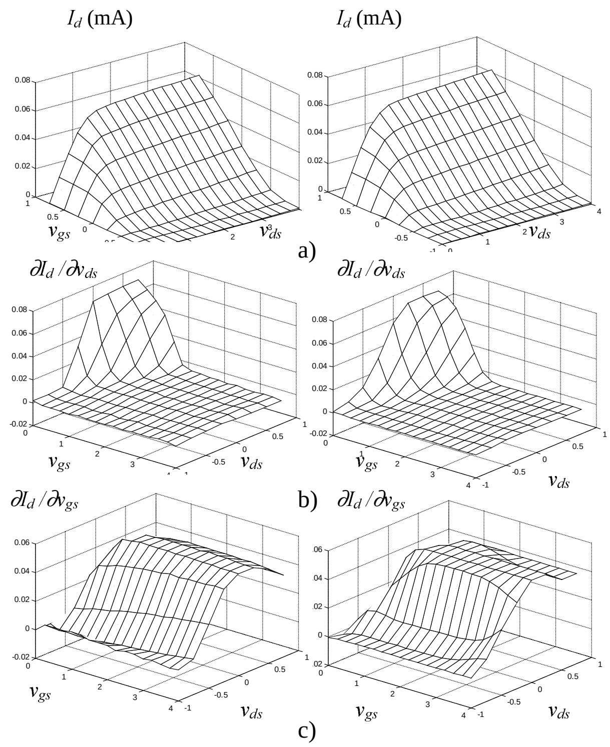

Figure 1. Measured functions (left) and approximation given by the SPWL model (right) for a) I_{ds}, b) I_{ds} derivative with respect to v_{gs}, and c) I_{ds} derivative wrt v_{ds}.

5. CONCLUSIONS

The SPWL model has been presented as a useful tool to model the large-signal behavior of MESFET/HEMT transistors. It allows getting a smooth and derivable approximation with a low number of parameters and with a low computational burden. Moreover, the smoothness varies monotonically with a single smoothing parameter of the function. In this way, the use of the SPWL model inherently provides some kind of regularization of the modeling problem. The proposed SPWL model has been applied to model a HEMT transistor under optical illumination using real measurements. When compared with neural networks-based alternatives the SPWL model shows clear advantages both in terms of performance and computational burden.

Model	No. Of Param	Function SNR (dB)	$\partial I_{ds} / \partial v_{ds}$ SNR (dB)	$\partial I_{ds} / \partial v_{gs}$ SNR (dB)
MLP(5)	36	25.50	14.00	14.40
CPWL(5)	36	30.62	9.52	13.04
SPWL(5)	**37**	**32.13**	**16.63**	**16.40**
MLP(9)	64	27.17	12.62	14.75
CPWL(10)	66	32.44	10.44	12.96
SPWL(10)	**67**	**34.35**	**18.40**	**17.15**
MLP(11)	78	28.09	13.71	14.66
CPWL(12)	78	33.04	11.51	14.34
SPWL(12)	**79**	**35.16**	**18.88**	**17.16**
GRBF(8)	88	28.84	10.09	13.79
Analyt. Model	98	32.92	-0.8674	16.53

Table I. Comparison of results for a Philips D02AH HEMT transistor. In the first column the number between parentheses indicates the number of components of the models.

6. REFERENCES

[1] S. Maas, *Nonlinear Microwave Circuits,* Norwood, MA: Artech House, 1988.

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

[5] K.Shirakawa, et al, "A large signal characterization of a HEMT using a multilayered neural network*", IEEE Trans. Microwave Theory Technol*., vol. 45, pp. 1630-1633, 1997.

[6] I. Santamaría, et al., " A nonlinear MESFET model for intermodulation analysis using a GRBF network", *Neurocomputing*, vol. 25, pp. 1-18, 1999.

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

[8] T. Poggio, F. Girosi, "Networks for approximation and learning", *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1481-1497, 1990.

[9] C. Navarro et al, "Large signal dynamic properties of GaAs MESFET/HEMT devices under optical illumination*", Proc. of the GAAS'98 Symposium*, pp. 350-353, Amsterdam, 1998.