IADIS International Journal on Computer Science and Information Systems
Vol. 4, No. 1, pp. 93-111
ISSN: 1646-3692

Efficient Electronic Device Modeling and Design Using Advanced CAD Tools

Said Gaoua [(1)], Farah A. Mohammadi [(2)], Mustapha C.E. Yagoub [(3)]

[(1)] Faculty of Electronics and Informatics, USTHB, Algiers, 16111 Algeria
[(2)] ECE Dept, Ryerson University, Toronto, ON, M5B 2K3 Canada
[(3)] SITE, University of Ottawa, Ottawa, ON, K1N 6N5 Canada.

Abstract

Because of the increasing demand for ever-higher level of electronic device integration and miniaturization, modern design requires massive computational tasks during simulation, optimization and statistical analyses, requiring robust modeling tools so that the whole process can be achieved reliably. In this paper, the authors developed advanced computer-aided design tools to efficiently model devices such as transistors and successfully predict the overall circuit performance. The proposed tools are demonstrated through examples.

Keywords: Fuzzy logic, neural networks, symbolic computation, transistors.

1 Introduction

Recent developments in modern communication market highlight an ever-increasing demand for more capability and functionality. Combined to other constraining factors like size and weight, the drive in the microelectronic industry for ever-higher integration and reliability leads to massive and highly repetitive computational tasks during simulation, optimization and statistical analyses, requiring that the models be permanently upgraded so that the design can be achieved accurately [1]-[3]. As such, there is a challenge for further research towards improvement of existing Computer-Aided Design (CAD) tools for high-frequency communication circuit modeling and design. Among all high-frequency active component models, unreliable transistor models can easily lead to an unsuccessful design due to their strong influence on the overall circuit behavior. Thus, the efficiency of such models in terms of accuracy and speediness is critical to assure a reliable design.

In the recent years, neural (NN) and fuzzy-neural networks (FNN) gained popularity as fast and flexible vehicle to high-frequency modeling, simulation and optimization [4]-[8]. Trained from measured/simulated data, fast and accurate neural models can be utilized in place of computationally intensive physics/EM models to speed-up the overall design process.

This work is supported in part by Natural Science and Engineering Research Council of Canada and in part by Canada Foundation for Innovation.

In this paper, the authors developed robust neural-based CAD tools to efficiently model high-frequency field effect transistors (FETs) and accurately predict active circuit responses. The proposed tools are demonstrated through examples.

2 Neural Network Review

A neural network (NN) is a model that has the ability to learn and generalize arbitrary continuous multi-dimensional input-output relationships. Let **x** be an N_x-vector $\{x_i,\ i = 1 \ldots N_x\}$ containing the inputs and **y** an N_y-vector $\{y_k,\ k = 1 \ldots N_y\}$ containing the outputs from the output neurons. The original problem can be expressed as **y** = f (**x**), while the neural network model for the problem is

$$\mathbf{y}_{\mathbf{NN}} = \tilde{\mathbf{y}}(\mathbf{x}, \mathbf{w}), \tag{1}$$

where **w** is a N_w-vector $\{w_i,\ i = 1 \ldots N_w\}$ containing all the weight parameters representing the connections in the neural network. The definition of **w** and the way in which $\mathbf{y}_{\mathbf{NN}}$ is computed from **x** and **w** determines the structure of the neural network.

2.1 MLP Neural Networks

The most commonly used neural network configuration is the Multi-Layer Perceptrons (MLP) where the neurons are grouped into layers as shown in Fig. 1. Layer L_1 is the input layer. Layers L_2 to L_{L-1} are called hidden layers, while the last layer L_L, the output layer, contains the response to be modeled.

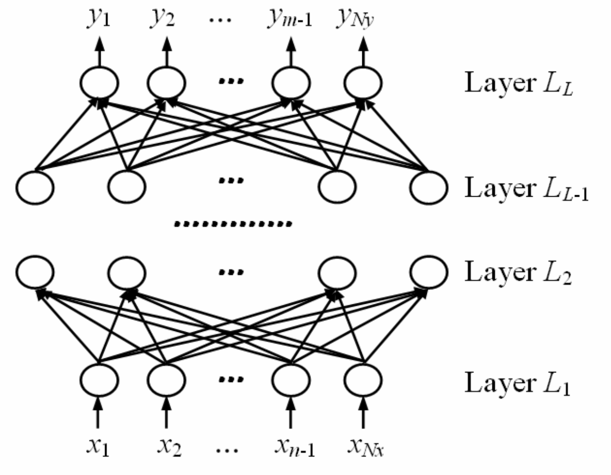

Fig. 1. Structure of the MLP neural network.

The layers are placed end to end with neuron connections between them. For such a network, the function in (1) is obtained on the basis of the layer of entry while using

$$z_i^1 = x_i,\ i = 1 \ldots N_x, N_x = N_1, \tag{2}$$

z_i^1 is the output of the i^{th} neuron of the input layer, and while proceeding layer by layer, the output at the end of layer L_l is given in the form of the activation function $\sigma(.)$ [4]

$$z_j^l = \sigma \left(\sum_{k=0}^{N_{l-1}} w_{jk}^l \; z_k^{l-1} + w_{j0}^l \right), \tag{3}$$

where $j = 1 \ldots N_l$ and $l = 2 \ldots L$, to reach the output layer that gives

$$y_k = z_k^L, \; k = 1 \ldots N_y, N_y = N_L, \tag{4}$$

In (2)-(4), N_l is the number of neurons in layer L_l, w_{jk} represents the weight of the connection between the k^{th} neuron of layer L_{l-1} and the j^{th} neuron of layer L_l.

2.2 KBNN Neural Networks

The modeling approach using MLP structures could generate accurate as well as fast models. However, since MLP is a kind of black-box model structurally embedding no-problem dependant information, the training process could necessitate a huge amount of data to efficiently learn the desired input/output relationships [4], [7], [8]. Generating large amounts of training data could be very expensive for high-frequency problems [4], [7], so existing knowledge can provide additional information to the original problem that may not be adequately represented by the limited training data. In Knowledge-Based Neural Networks (KBNN), the neural network can help bridge the gap between empirical model and desired solutions. The structure of KBNN [4], illustrated in Fig. 2, is constituted of six layers, which are not fully connected to each other, namely input layer *X*, knowledge layer *Z*, boundary layer *B*, region layer *R*, normalized region layer *R'* and output layer *Y*. The knowledge layer *Z* is the place where microwave knowledge resides in the form of single or multidimensional function $\psi(.)$. For knowledge neuron *i* in the *Z* layer [4]

$$z_i = \psi_i(\mathbf{x}, \mathbf{w}_i), \quad i = 1 \ldots N_z \tag{5}$$

where **x** is the neural network inputs vector, N_z is the number of knowledge neurons, and $\mathbf{w}_i$ is a vector of parameters in the knowledge formula. The knowledge function $\psi_i(\mathbf{x}, \mathbf{w}_i)$ is usually in the form of empirical or semi-analytical functions. The boundary layer *B* can incorporate knowledge in the form of problem dependent boundary functions *B*(.). Neuron *i* in the layer *B* is calculated by

$$b_i = B_i(\mathbf{x}, \mathbf{v}_i), \quad i = 1 \ldots N_b \tag{6}$$

where $\mathbf{v}_i$ is a vector of the parameters in B_i defining an open or closed boundary in the input space **x**. The region layer *R* contains neurons to construct regions from boundary neurons,

$$r_i = \prod_{j=1}^{N_b} \sigma(\alpha_{ij} b_j + \theta_{ij}), \quad i = 1 \ldots N_r \tag{7}$$

where α_{ij} and θ_{ij} are the scaling and bias parameters, respectively. The normalized region layer *R'* contains rational function-based neurons to normalize the outputs of region layer,

$$r_i' = \frac{r_i}{\sum_{j=1}^{N_r} r_j}, \quad i = 1 \ldots N_{r'} \quad \text{where } N_r = N_{r'} \tag{8}$$

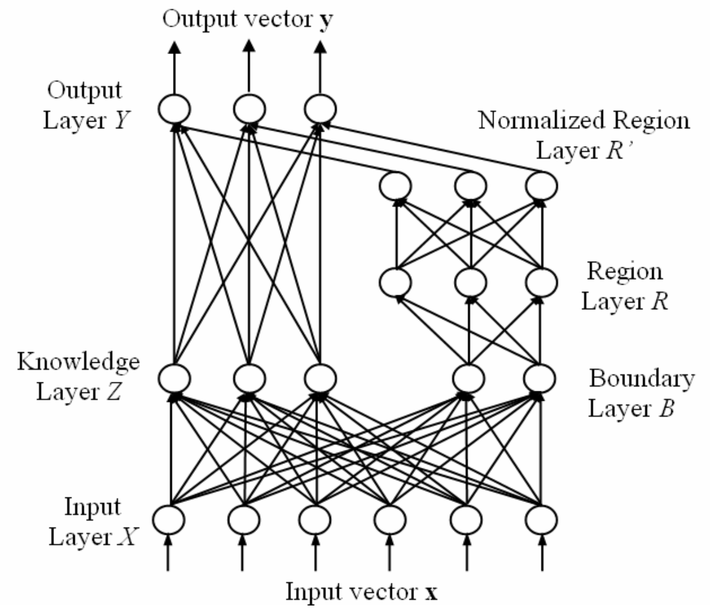

Fig. 2. Structure of the KBNN neural network.

The output layer *Y* combines knowledge neurons and normalized region neurons

$$y_i = \sum_{i=1}^{N_z} \beta_{ji} z_i (\sum_{k=1}^{N_{r'}} \rho_{jik} r_k') + \beta_{j0}, \quad j = 1 \ldots N_y \tag{9}$$

where β_{ji} reflects the contribution of the i^{th} knowledge neurons to output neuron y_i and β_{j0} is the bias parameter, ρ_{jik} is one indicating that region r_k' is the effective region of the i^{th} knowledge neuron contributing to the j^{th} output. A total of $N_{r'}$ regions are shared by all the output neurons. Compare to MLP structures, the prior knowledge in KBNN gives neural network more information about the original microwave problem, besides the information included in the training data. Consequently, KBNN models have better reliability when training data is limited or when the model is used beyond training range [4].

2.3 PKI Neural Networks

Similarly to KBNN, the Prior Knowledge Input (PKI) neural structure could complement the capability of learning and generalizing of the neural network [4]. The structure in Fig. 3 uses an empirical model as the prior knowledge part and a neural network to map between the inputs of the original problem, outputs of the empirical model and outputs of the neural model. For each set of input vector $\mathbf{x_k}$ in the training data, a corresponding vector $\mathbf{\hat{y}_k}$ is computed using the empirical model. The neural network will then learn the relationship between original problem inputs, empirical model outputs ($\mathbf{x_k}$, $\mathbf{\hat{y}_k}$), and the desired output vector $\mathbf{y_k}$. Compare to MLP, the outputs of the empirical model help getting better accuracy.

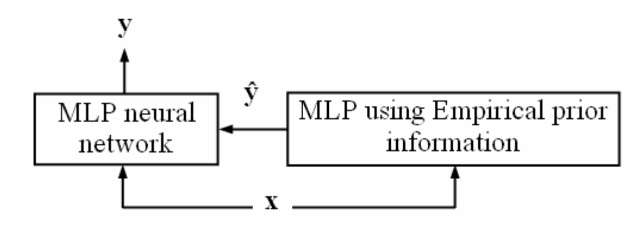

Fig. 3. Structure of the PKI neural network.

2.4 Fuzzy-Neural Networks

The combination of fuzzy systems and neural networks can significantly improve the learning ability of a model, especially when the solution is not unique or in presence of uncertainties/noise in data used in model training. It is the case when designers use an electrical equivalent circuit to characterize a transistor behavior, this circuit is not unique but strongly dependent on the technology, the operating frequency, and the accuracy of the input measured data [9]. Neural network training algorithms like back-propagation are usually gradient based learning procedure and the trained network can be used to classify data by extrapoling the information learned, while fuzzy clustering involves an unsupervised learning approach [10]-[12].

Among existing fuzzy methods, the Fuzzy c-means (FCM) method is an iterative data clustering technique wherein each data point belongs to a cluster to some degree that is specified by a membership grade [11]. Clustering in N unlabeled data $\mathbf{x} = \{x_i, i = 1 \dots N\}$ is the assignment of c number of partition labels to the vectors in X. The clustering problem is to find the optimum matrix $\mathbf{U} = [u_{ij} \in [0, 1], i = 1 \dots c; j = 1 \dots N]$ which minimizes the within cluster sum of distances J_h defined as [11], [12]

$$J_h(U, v) = \sum_{k=1}^{N} \sum_{i=1}^{c} (u_{ik})^h \|x_k - v_i\|^2 \tag{10}$$

where h controls the degree of fuzziness, u_{ik} describes the belongness of x_i to cluster k,

$$u_{ik} = \left(\sum_{j=1}^{c} \left[\frac{\|x_k - v_i\|}{\|x_k - v_j\|} \right]^{\frac{2}{h-1}} \right)^{-1} \tag{11}$$

and v_i is the centroid of i^{th} cluster,

$$v_i = \left(\sum_{k=1}^{N} (u_{ik})^h \, x_k \right) \left(\sum_{k=1}^{N} (u_{ik})^h \right)^{-1} \tag{12}$$

In (11), with larger values of h, the membership u_{ik} tends to be closer to $1/c$, and therefore fuzzier [12].

3 Transistor Modeling

Because the FET is one of the most widely used active devices in modern communication systems, a large number of modeling approaches have been proposed to characterize its behavior [13]-[18]. Detailed physics-based transistor models are accurate but slow. Table look-up models can be fast, but suffer from the disadvantages of large memory requirements and limitations on number of parameters.

Nevertheless they are difficult to develop, equivalent circuit models remain the most used modeling approach where the element values can be determined either by direct extraction [13] or by optimization-based extraction [14]. Fast and simple to implement, direct-extraction techniques provide adequate values for the more dominant circuit model elements but they cannot determine all the extrinsic elements uniquely [15]. On the other side, optimization-based extraction techniques are more accurate but computationally intensive and relatively sensitive to the choice of starting values. Also, to make them attractive to non-experienced users, such techniques often assume a *prior universal* topology referred as the *FET standard topology* or circuit #1 (Fig. 4) [16].

Determining the most suitable small-signal equivalent circuit topology and accurately extracting its element parameters is the aim of the proposed approach. Based on a large literature review, the authors created a library with different circuit topologies displayed in Fig. 5 to 8 [17]-[20].

In electronics, any network with external connections (ports) can be described with a set of independent and dependent variables. The independent variables represent external inputs while the dependent variables represent the system responses. Utilizing circuit theory transformations [21], [22], different sets of network parameters can be used to mathematically describe the input-output relationships such as *Z*-parameters (impedance matrix), *Y*-parameters (admittance matrix), *T*-parameters (transfer matrix), or *S*-parameters (scattering matrix). There are several advantages of using *S*- and *T*-parameters in high frequencies; *S*-parameters are the most reliable in terms of measurements, while *T*-parameters are the most convenient for computing the overall characteristic matrix of a combination of cascaded high-frequency networks.

For a given transistor, a standard topology extraction [13] was then performed and the obtained *S*-parameters ($S_{ij}^{\mathbf{s}}$, $i, j = 1, 2$) from the standard topology were compared to the measured *S*-parameters (denoted as $S_{ij}^{\mathbf{m}}$, $i, j = 1, 2$). If the difference is greater than the user-defined error, a new circuit topology should be selected from the library. By combining the Fuzzy c-means method and the small-signal representation of the device behavior, the most suitable transistor topology can be deduced, as shown in Fig. 9.

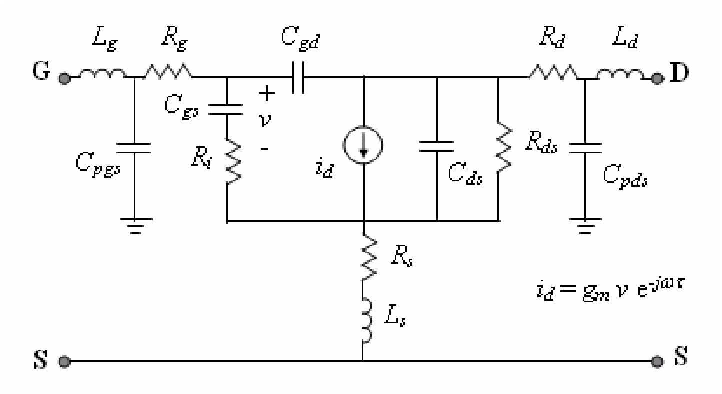

Fig. 4. FET standard circuit (topology # 1) [16].

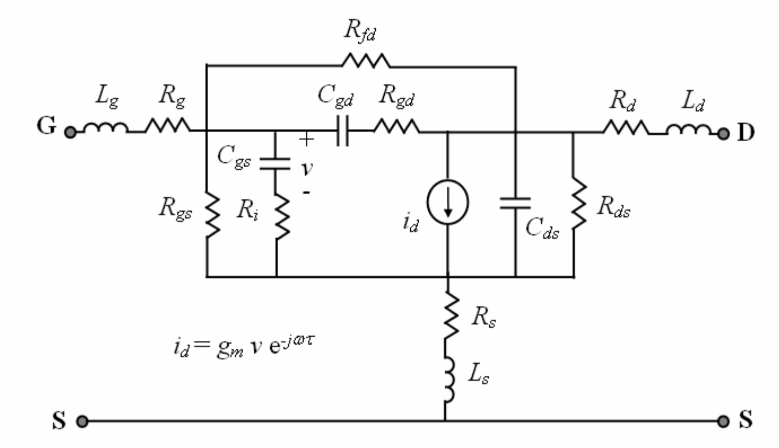

Fig. 5. FET circuit topology #2 as reported in [17].

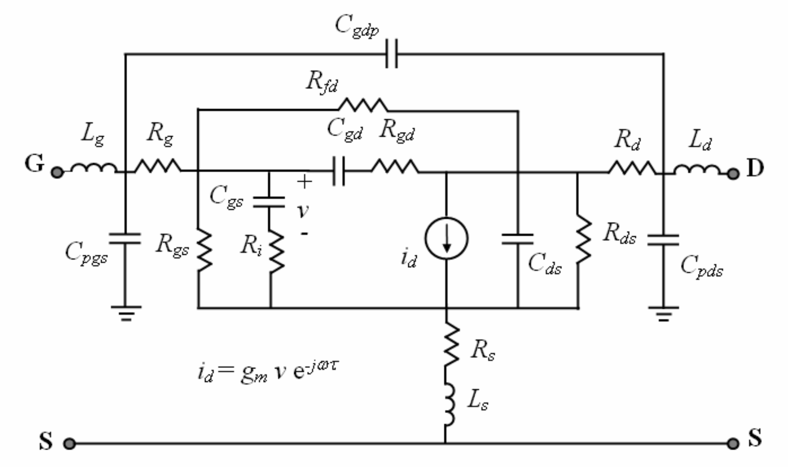

Fig. 6. FET circuit topology #3 as reported in [18].

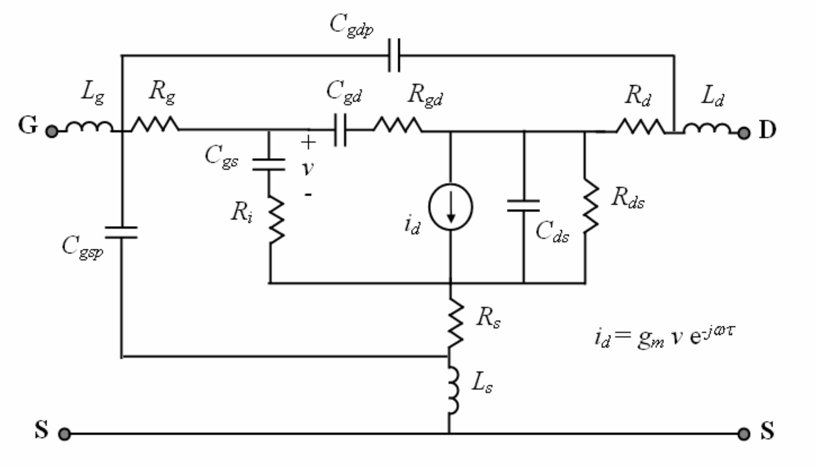

Fig. 7. FET circuit topology #4 as reported in [19].

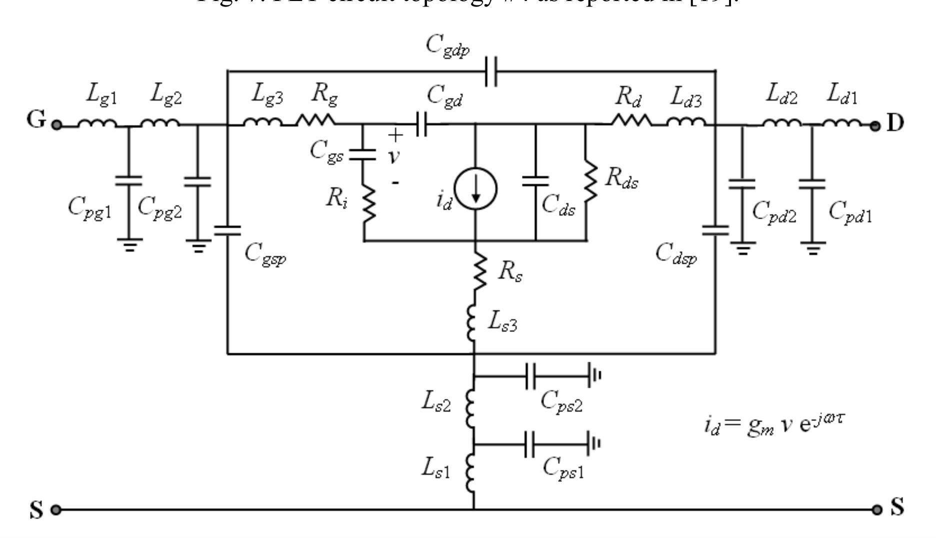

Fig. 8. FET circuit topology #5 as reported in [20].

In fact, for any circuit #*k* of the library, the related $\mathbf{S}^{\mathbf{k}}$ matrix was compared to the given input $\mathbf{S}^{\mathbf{m}}$ matrix and each element of the 2x2 error matrices $\mathbf{E}^{\mathbf{k},\,\mathbf{Re}}$ and $\mathbf{E}^{\mathbf{k},\,\mathbf{Im}}$,

$$E_{ij}^{\mathbf{k},\mathbf{Re}} = \mathrm{Re}\left(S_{ij}^{\mathbf{k}} - S_{ij}^{\mathbf{m}}\right) \quad i,j = 1,2 \tag{13}$$

$$E_{ij}^{\mathbf{k},\mathbf{Im}} = \mathrm{Im}\left(S_{ij}^{\mathbf{k}} - S_{ij}^{\mathbf{m}}\right) \quad i,j = 1,2 \tag{14}$$

will receive a score depending on its value. Thus, topology #*k* with smallest $\mathbf{E}^{\mathbf{k},\,\mathbf{m}}$,

$$E^{\mathbf{k},\mathbf{m}} = \sum_{i=1}^{2}\sum_{j=1}^{2}\left\{\left[\mathrm{Re}\left(S_{ij}^{\mathbf{k}} - S_{ij}^{\mathbf{m}}\right)\right]^2 + \left[\mathrm{Im}\left(S_{ij}^{\mathbf{k}} - S_{ij}^{\mathbf{m}}\right)\right]^2\right\} \tag{15}$$

i.e., smallest score, will be selected as the most suitable equivalent model topology. Here, Re(*) and Im(*) denote real part and imaginary part respectively. If the final error is not acceptable, a new circuit topology should be selected from the library.

By combining the FCM method and the small-signal neural representation of the device, our approach allowed selection of the most suitable transistor topology. Finally, MLP-NN models have been generated (one for each topology) and the selected topology was used to extract the corresponding circuit element values via a simple extraction loop. However, since there is *no prior* knowledge on the input *S*-parameters, it is impossible to compute numerically (15). Let $\{\Omega^{\mathrm{s}}\}$ be the set of P_s elements $\Omega^{\mathrm{s}}{}_p$ (p = 1 … P_s) in the standard topology. A symbolic code was developed using [23] to analytically derive the following nonlinear functions

$$S^{\mathbf{k}}_{ij} = f^{\mathbf{k}}_{ij}\left(S^{\mathbf{s}}_{ij}, \left\{\Omega^{\mathbf{k}}\right\}\right) \quad i,j=1,2 \quad k=1\ldots5 \tag{16}$$

where $\{\Omega^{\mathbf{k}}\}$ is the set of the P_k elements added in circuit #k versus the standard topology, e.g.,

$$\left\{\Omega^2\right\}_{FET} = \left\{R_{fd}, R_{gd}, R_{gs}\right\}\Big|_{C_{pgs}=C_{pds}=0}$$

$$\left\{\Omega^3\right\}_{FET} = \left\{R_{fd}, R_{gd}, R_{gs}, C_{gdp}\right\} \;,$$

$$\left\{\Omega^4\right\}_{FET} = \left\{R_{gd}, C_{gdp}, C_{gsp}\right\}\Big|_{C_{pgs}=C_{pds}=0}$$

$$\left\{\Omega^5\right\}_{FET} = \left\{\begin{matrix} C_{gdp}, C_{gsp}, C_{dsp}, C_{pg1}, C_{pg2}, C_{ps1}, C_{ps2}, \\ C_{pd1}, C_{pd2}, L_{g1}, L_{g2}, L_{s1}, L_{s2}, L_{d1}, L_{d2} \end{matrix}\right\}\Bigg|_{C_{pgs}=C_{pds}=0}$$

As an illustration, the procedure used to obtain the overall **S** matrix of the standard topology is presented in Appendix I along with a symbolic code to invert complex matrices. In fact, since the *Z*-matrix is the inverse of the *Y*-matrix, such procedure involves many matrix inversions. Therefore, the following alternative fuzzy criteria can be defined for each topology #k as

$$E^{\mathbf{k,s}} = \sum_{i=1}^{2}\sum_{j=1}^{2}\left\{\left[\mathrm{Re}\left(S^{\mathbf{k}}_{ij} - S^{\mathbf{s}}_{ij}\right)\right]^2 + \left[\mathrm{Im}\left(S^{\mathbf{k}}_{ij} - S^{\mathbf{s}}_{ij}\right)\right]^2\right\} \tag{17}$$

Since these equations are strongly interdependent and highly nonlinear, we used neural networks to learn them. By varying the values of the different elements $\Omega^{\mathrm{k}}{}_p$ (p = 1 … P_k). of set $\{\Omega^{\mathrm{k}}\}$, we can compute the $\mathbf{S}^{\mathbf{k}}$ scattering matrix and therefore, the difference $\{\mathbf{S}^{\mathbf{k}} - \mathbf{S}^{\mathbf{s}}\}$. As shown in Fig. 10, the resulting data in the form of

$$Tr^{\mathbf{k}} = \left[\underbrace{\mathrm{Re}\left(S^{\mathbf{k}}_{ij} - S^{\mathbf{s}}_{ij}\right), \mathrm{Im}\left(S^{\mathbf{k}}_{ij} - S^{\mathbf{s}}_{ij}\right)}_{8\text{ inputs }(i,j=1,2)}, \underbrace{\Omega^{\mathbf{k}}_1, \ldots, \Omega^{\mathbf{k}}_{p_{\mathbf{k}}}}_{p_{\mathbf{k}}\text{ outputs}}\right] \tag{18}$$

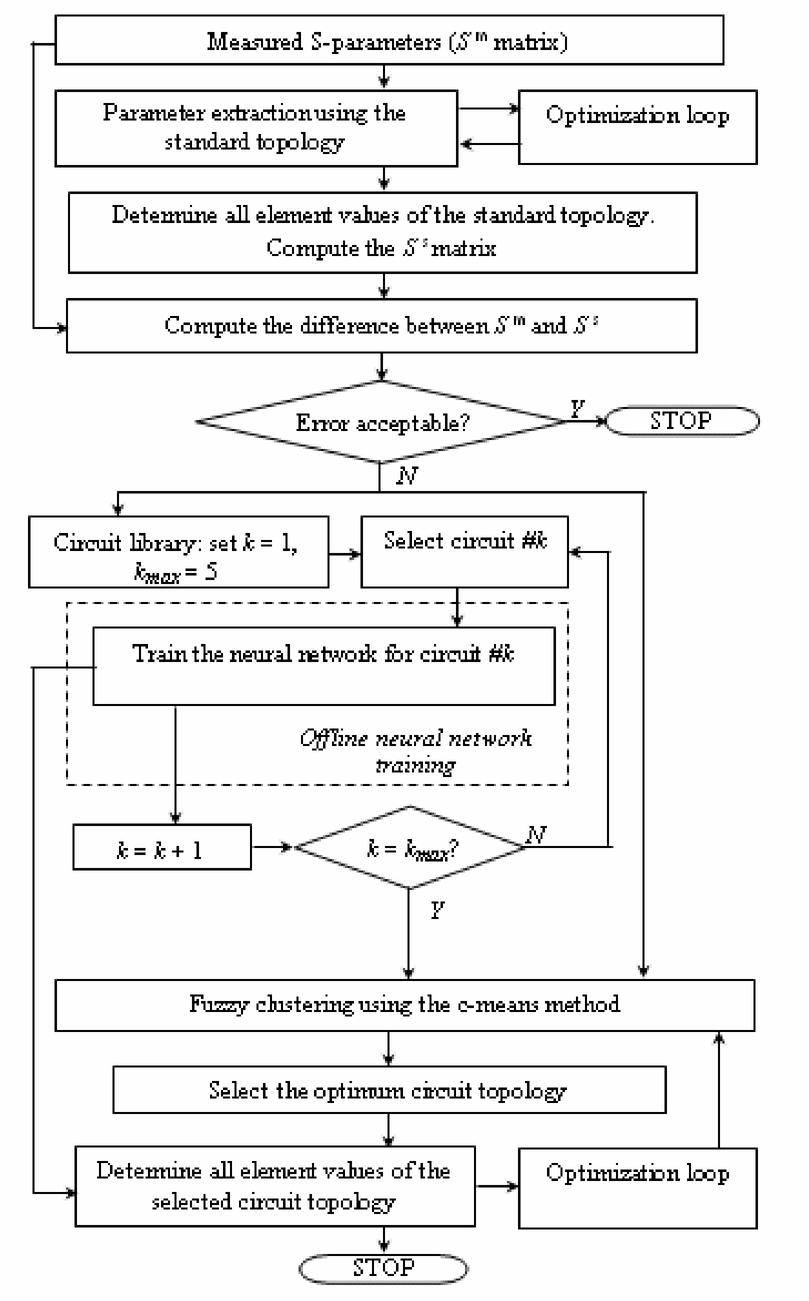
Measured S-parameters (S^m matrix)
Parameter extraction using the standard topology
Optimization loop
Determine all element values of the standard topology. Compute the S^s matrix
Compute the difference between S^m and S^s
Error acceptable?
Y
STOP
N
Circuit library: set k = 1, k_{max} = 5
Select circuit #k
Train the neural network for circuit #k
Offline neural network training
k = k + 1
k = k_{max}?
N
Y
Fuzzy clustering using the c-means method
Select the optimum circuit topology
Determine all element values of the selected circuit topology
Optimization loop
STOP

Fig. 9. Algorithm of the proposed method.

was submitted to a three-layer (MLP3) neural network structure for training using [24]. The input layer has 9 neurons (the 4 real and 4 imaginary parts in (18) and the operating frequency *f*) while the output layer contains P_k neurons. The hidden layer is composed of 22 to 45 neurons depending on the data file under training.

A final extraction is then performed in the form of an optimization loop with vector

$$\Omega = \left[\Omega_1^{\mathbf{k}}, \ldots, \Omega_{P_{\mathbf{k}}}^{\mathbf{k}}, \Omega_1^{\mathbf{s}}, \ldots, \Omega_{P_{\mathbf{s}}}^{\mathbf{s}}\right] \tag{19}$$

as starting vector. Since this vector is close to the final solution, it assures a very fast convergence. In fact, the maximum number of iterations for 100 sets of *S*-parameters did not exceed ten iterations with a CPU time of 11s and a user predefined error of 2%.

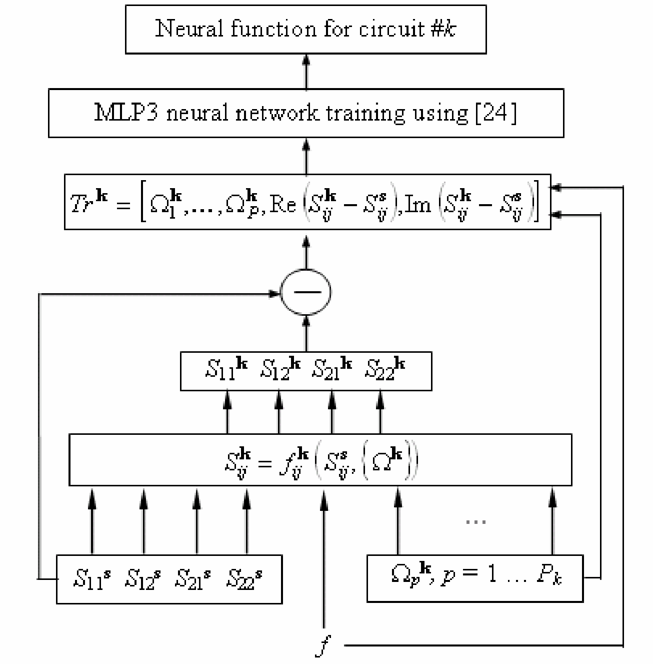
Neural function for circuit #k
MLP3 neural network training using [24]
$Tr^{\mathbf{k}} = \left[\Omega_1^{\mathbf{k}}, \ldots, \Omega_P^{\mathbf{k}}, \mathrm{Re}\left(S_{ij}^{\mathbf{k}} - S_{ij}^{s}\right), \mathrm{Im}\left(S_{ij}^{\mathbf{k}} - S_{ij}^{s}\right)\right]$
—
$S_{11}^{\mathbf{k}}$ $S_{12}^{\mathbf{k}}$ $S_{21}^{\mathbf{k}}$ $S_{22}^{\mathbf{k}}$
$S_{ij}^{\mathbf{k}} = f_{ij}^{\mathbf{k}}\left(S_{ij}^{s}, \left(\Omega^{\mathbf{k}}\right)\right)$
...
S_{11}^{s} S_{12}^{s} S_{21}^{s} S_{22}^{s}
$\Omega_p^{\mathbf{k}}, p = 1 \ldots P_k$
f

Fig. 10. Neural network model development for circuit #*k*.

4 Transistor Characterization

4.1 Example 1: MESFET with Simulated Data

The first device to be characterized is the MESFET reported in [25] using FET topology #4. Since in this paper all circuit element values are given as well as the final error between measured and calculated *S*-parameters, a reliable comparison can be performed for a full validation. In fact, by comparing the S-parameters (Fig. 11) and the extracted values given in [25] with those obtained in 2.3 seconds using our technique (Table I), topology #4 achieved the closest agreement with a smaller final error (2.9% vs. the 8.4% in [25]) defined for a set of N_f selected frequency values fq ($q = 1 \ldots N_f$) as [25]

$$E^{\mathbf{k},\mathbf{m}} = \sum_{q=1}^{N_f} \sum_{i=1}^{2} \sum_{j=1}^{2} \left| 1 - \frac{S_{ij}^{\mathbf{k}}\left(f_q\right)}{S_{ij}^{\mathbf{m}}\left(f_q\right)} \right|^2 \tag{20}$$

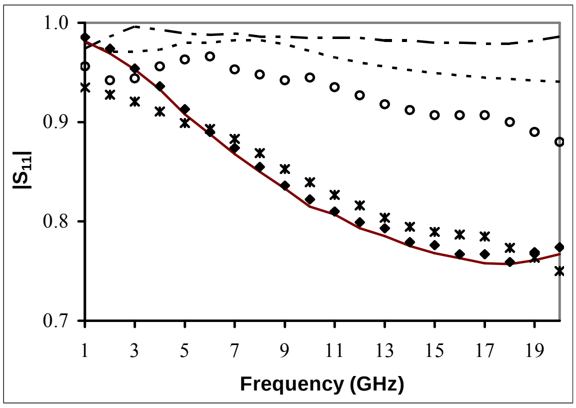

Fig. 11. MESFET: Comparison of S_{11} parameter given in [25] (♦) with those extracted using: ---- : standard topology, —— : topology #2, **o** : topology #3, —— : topology #4, * : topology #5.

Table I. MESFET: Comparison Between the Parameters Reported in [25] and our computed results.

	Circuit #4	Our Values
C_{gs} (pF)	0.277	0.215
C_{gd} (pF)	0.0207	0.0211
C_{ds} (pF)	0.0993	0.101
g_m (mS)	26.9	27.3
τ (ps)	1.22	1.25
R_i (Ω)	15.3	15.1
R_{gd} (Ω)	43.8	43.6
R_{ds} (Ω)	215	218
R_g (Ω)	8.9	9.1
R_s (Ω)	7.5	7.3
R_d (Ω)	13.6	13.2
L_s (nH)	0.437	0.441
L_d (nH)	0.452	0.447
L_g (nH)	0.254	0.258
C_{gsp} (pF)	0.0409	0.0397
C_{gdp} (pF)	0.001	0.001
Error (%)	8.4	2.9

4.2 Example 2: PHEMT with Measured Data

In a second example, we measured the *S*-parameters of an AlGaAs/InGaAs-GaAs PHEMT. Then, we specified a user-defined error of 2%. After 2.1 seconds, our method showed that FET topology #3 was the most appropriate (Fig. 12) with an acceptable final error of 1.8%, smaller than the above user-defined error.

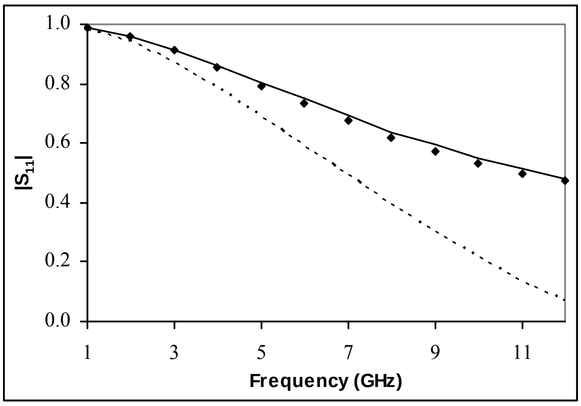

Fig. 12. PHEMT: Comparison of measured S_{11} parameter (♦) with those extracted using: ---- : standard topology, —— : topology #3.

5 Circuit Modeling and Design

For circuit design, we trained a PKI structure for a one-stage amplifier to learn its input-output relationship and thus, to predict the two-stage amplifier response. The PKI input vector contained the input power, the DC bias, and the frequency. The output vector contained the output power of the two first harmonics. Data generation was performed from 0.5GHz to 1.5GHz, step size of 0.025GHz, while the DC voltage was varied from 2V to 4 V, step size of 0.2V. The input power was swept from -100dBm to -90dBm, step size of 1dBm. A KBNN was also built using the same data range to enhance the two-stage model prediction beyond the training range. The empirical coarse model was given from an MLP model using the same data. As expected, the PKI model allowed significant reduction of the CPU time (0.2s vs. 12s for the original simulation run in ADS [20]). Furthermore, and as expected, KBNN showed a better agreement with original data from [20] than those given by the MLP for input values beyond the training range (Table II).

Table II. Two-Stage Amplifier: Fundamental Output Power {Pout (ω)} and Second Harmonic Output Power {Pout (2ω)}.

	$P_{in}(\omega)$=-95dBm		$P_{in}(\omega)$=-85dBm	
	$P_{out}(\omega)$ in dBm	$P_{out}(2\omega)$ in dBm	$P_{out}(\omega)$ in dBm	$P_{out}(2\omega)$ in dBm
[20]	-76.67	-191.05	-62.81	-168.24
MLP	-70.45	-159.91	-81.35	-187.46
KBNN	-75.44	-189.20	-60.97	-165.29
PKI	-76.67	-190.31	-57.41	-160.22

6 Conclusions and Future Work

In this paper, different advanced CAD tools have been presented. They combine fuzzy and neural techniques to obtain the most suitable small-signal electrical equivalent circuit of a given transistor and efficiently predict a circuit response. They will be shortly extended to include nonlinear as well as thermal transistor behaviors at the component level, and highly nonlinear circuits at the circuit level.

References

[1] T. Edwards, Countdown to the microwave millennium, *Microwave J.*, **41** (1998), 70-81.

[2] K.D. Cornett, A wireless R&D perspective on RF/IF passives integration, *IEEE Bipolar/BiCMOS Circuits Tech. Meeting*, 2000, Minneapolis, MN, 187-190.

[3] R.R. Tummala, M. Swaminathan, M.M. Tentzeris, J. Laskar, G.-K. Chang, S. Sitaraman, D. Keezer, D. Guidotti, Z. Huang, K. Lim, L. Wan, S.K. Bhattacharya, V. Sundaram, F. Liu, P. M. Raj, The SOP for miniaturized, mixed-signal computing, communication, and consumer systems of the next decade, *IEEE Trans. Advanced Packaging*, **27** (2004), 250-267.

[4] Q.J. Zhang, K.C. Gupta, *Neural Networks for RF and Microwave Design*, Artech House, Norwood (2000).

[5] M.C.E. Yagoub, Optimisation des performances de modules multi-puces. Modélisation par réseaux de neurones, *Annales des Télécommunications*, **59** (2004), 941-966.

[6] J. Bandler, M.A. Ismail, J.E Rayas-Sanchez, Q.J. Zhang, New directions in model development for RF/microwave components utilizing artificial neural networks and space mapping, *IEEE Int. Antenna Prop. Symp.*, 1999, Orlando, FL, 2572-2575.

[7] V.K. Devabhaktuni, B. Chattaraj, M.C.E. Yagoub, Q.J. Zhang, Advanced microwave modeling framework exploiting automatic model generation, knowledge neural networks and space mapping, *IEEE Int. Microwave Theory Tech. Symp.*, 2002, Seattle, WA, 1098-1100.

[8] P.M. Watson, K.C. Gupta, R.L. Mahajan, Development of knowledge based artificial neural network models for microwave components, *IEEE Int. Microwave Theory Tech. Symp.*, 1998, Baltimore, MD, 9-12.

[9] S. Gaoua, F.A. Mohammadi, M.C.E. Yagoub, "Fuzzy-neural CAD tools for advanced communication system design," *IEEE Information and Communication Technologies Int. Symp.*, 2007, Fez, Morocco, B9.1_1-B9.1_4.

[10] B. Karlik, H. Torpi, M. Alci, A fuzzy-neural approach for the characterisation of the active microwave devices, *Int. Conf. Microwave Telecommunication Technology*, 2002, Sevastopol, Ukraine, 114-117.

[11] C. Fager, L.J.P. Linner, J.C. Pedro, Optimal parameter extraction and uncertainty estimation in intrinsic FET small-signal models, *IEEE Trans. Microwave Theory Tech.*, **50** (2002), 2797-2803.

[12] Y. Hu, K. Ashenayi, R. Veltri, G. O'Dowd, G. Miller, R. Hurst, R. Bonner, A comparison of neural network and fuzzy c-means methods in bladder cancer cell

classification, *IEEE International Conf. on Neural Networks*, 1994, Orlando, FL, 3461-3466.

[13] J.M. Golio, *Microwave MESFETs and HEMTs*, Artech House, Boston (1991).

[14] L. Fujiang, G. Kompa, FET model parameter extraction based on optimization with multiplane data-fitting and bidirectional search-a new concept, *IEEE Trans. Microwave Theory Tech.*, **42**, 1994, 1114-1121.

[15] C. Van Niekerk, P. Meyer, D.M.M.-P. Schreurs, P.B. Winson, A robust integrated multibias parameter-extraction method for MESFET and HEMT models, *IEEE Trans. Microwave Theory Tech.*, **48**, 2000, 777-786.

[16] G. Dambrine, A. Cappy, F. Heliodore, E. Playez, A new method for determining the FET small-signal equivalent circuit, *IEEE Trans. Microwave Theory Tech.*, **36**, 1988, 1151-1159.

[17] R. Menozzi, A. Piazzi, F. Contini, Small-signal modeling for microwave FET linear circuits based on a genetic algorithm, *IEEE Trans. Circuits and Systems*, **43**, 1996, 839-847.

[18] M. Fernandez-Barciela, P.J. Tasker, Y. Campos-Roca, M. Demmler, H. Massler, E. Sanchez, M.C. Curras-Francos, M. Schlechtweg, A simplified broad-band large-signal nonquasi-static table-based FET model, *IEEE Trans. Microwave Theory Tech.*, **48**, 2000, 395-405.

[19] R. Anholt, S. Swirhun, Equivalent-circuit parameter extraction for cold GaAs MESFET's, *IEEE Trans. Microwave Theory Tech.*, **39**, 1991, 1243-1247.

[20] ADS 2006, *Agilent Technologies*, Palo Alto, CA, USA.

[21] K. Kurokawa, "Power waves and the scattering matrix," *IEEE Trans. Microwave Theory Tech.*, **13** (1965), 194-202.

[22] D.A. Frickey, "Conversions between S, Z, Y, h, ABCD, and T parameters which are valid for complex source and load impedances," *IEEE Trans. Microwave Theory Tech.*, **42** (1994), 205-211.

[23] Maple 8 reference manual, New-York, 2001.

[24] NeuroModeler 1.2, *Carleton University*, Ottawa, ON, Canada, 2000.

[25] M.K. Ahmed, S.M.M. Ibrahem, Small signal GaAs MESFET model parameters extracted from measured S-parameters, *National Radio Science Conf.*, 1996, Cairo, Egypt, 507-515.

APPENDIX I – S MATRIX OF TOPOLOGY #1

Let us divide the elements of the transistor equivalent circuit (topology #1, Fig. 4) into two groups: intrinsic and extrinsic elements, as shown in Fig. 13 [16]. The intrinsic elements are those in the transistor chip while the extrinsic ones (outside the dotted rectangle) represent the parasitic elements.

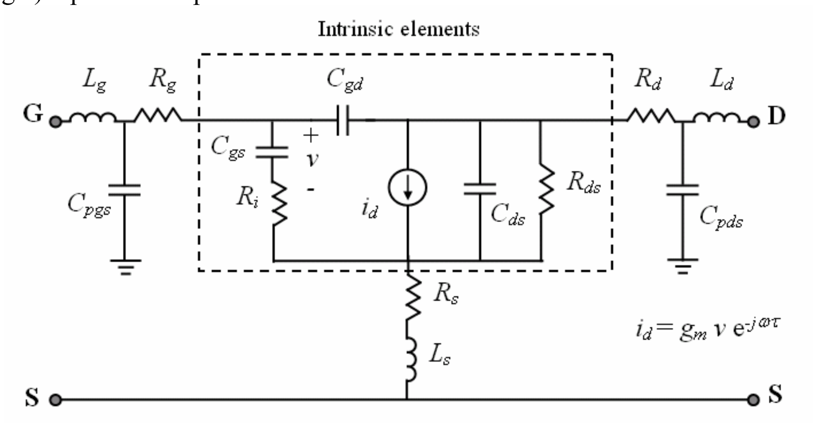

Fig. 13. Decomposition of the circuit into two parts: intrinsic and extrinsic [16].

Transistor Intrinsic Admittance Matrix

Accordingly to the circuit topology, the elements of the intrinsic admittance matrix, denoted $\mathbf{Y}_{\mathbf{int}}$, can formulated as [16]

$$Y_{11\ \mathrm{int}} = \frac{R_i C_{gs}^2 \omega^2}{1+\omega^2 C_{gs}^2 R_i^2} + j\omega\left(\frac{C_{gs}}{1+\omega^2 C_{gs}^2 R_i^2} + C_{gd}\right) \qquad \text{(AI-1)}$$

$$Y_{12\ \mathrm{int}} = -j\omega C_{gd} \qquad \text{(AI-2)}$$

$$Y_{21\ \mathrm{int}} = \frac{g_m \exp(-j\,\omega\,\tau)}{1+jR_i\,C_{gs}\,\omega} - j\omega C_{gd} \qquad \text{(AI-3)}$$

$$Y_{22\ \mathrm{int}} = g_d + j\omega\left(C_{ds} + C_{gd}\right) \qquad \text{(AI-4)}$$

Transistor Overall S Matrix

To obtain the overall **S** matrix from $\mathbf{Y}_{\mathbf{int}}$, some matrix manipulations are required as described in Fig. 14. Since this process involves many **Z** to **Y** matrix inversions (since $\mathbf{Y} = \mathbf{Z}^{-1}$), we developed a symbolic code to address such issues.

RESULTS FROM THE SYMBOLIC CODE: FINAL Z MATRIX

$$a11 := -rb\ cbc\ cc\ w$$

$$a12 := I\,(cc + cbc)$$

$$a13 := \frac{rl}{1 + I\,w\,cl\,rl}$$

$$a14 := \frac{1 + I\,w\,cp\,rpi}{rpi}$$

$$a1 := \frac{I\,rb\,cc}{-rb\ cbc\ cc\ w + I\,(cc + cbc)}$$

$$a2 := \frac{1}{-rb\ cbc\ cc\ w^{2} + I\,(cc + cbc)\,w}$$

$$a3 := \frac{I\,rb\,cbc}{-rb\ cbc\ cc\ w + I\,(cc + cbc)}$$

$$a4 := \frac{rpi}{1 + I\,w\,cp\,rpi}$$

$$zz := \begin{bmatrix} \frac{y_{[2,\,2]}}{\%1} & -\frac{y_{[1,\,2]}}{\%1} \\ -\frac{y_{[2,\,1]}}{\%1} & \frac{y_{[1,\,1]}}{\%1} \end{bmatrix}$$

$$\%1 := y_{[1,\,1]}\,y_{[2,\,2]} - y_{[1,\,2]}\,y_{[2,\,1]}$$

$$z_{[1,\,1]} := \frac{y_{[2,\,2]}}{y_{[1,\,1]}\,y_{[2,\,2]} - y_{[1,\,2]}\,y_{[2,\,1]}} + \frac{rl}{1 + I\,w\,cl\,rl} + rc + \frac{I\,rb\,cc}{-rb\ cbc\ cc\ w + I\,(cc + cbc)} + I\,w\,(le + lb)$$

$$z_{[2,\,2]} := \frac{y_{[1,\,1]}}{y_{[1,\,1]}\,y_{[2,\,2]} - y_{[1,\,2]}\,y_{[2,\,1]}} + rc + re + I\,w\,(lc + le)$$

$$z_{[2,\,1]} := -\frac{y_{[2,\,1]}}{y_{[1,\,1]}y_{[2,\,2]}-y_{[1,\,2]}y_{[2,\,1]}}+re+I\,w\,le$$

$$z_{[1,\,2]} := -\frac{y_{[1,\,2]}}{y_{[1,\,1]}y_{[2,\,2]}-y_{[1,\,2]}y_{[2,\,1]}}+re+I\,w\,le$$

$$bl1 := -\frac{le}{lc}$$

$$bl2 := \frac{1}{-\frac{y_{[1,\,2]}}{y_{[1,\,1]}y_{[2,\,2]}-y_{[1,\,2]}y_{[2,\,1]}}+\frac{y_{[2,\,1]}}{y_{[1,\,1]}y_{[2,\,2]}-y_{[1,\,2]}y_{[2,\,1]}}}$$

$$bl3 := -\frac{y_{[1,\,2]}}{y_{[1,\,1]}y_{[2,\,2]}-y_{[1,\,2]}y_{[2,\,1]}}+\frac{y_{[2,\,1]}}{y_{[1,\,1]}y_{[2,\,2]}-y_{[1,\,2]}y_{[2,\,1]}}$$

$$bl4 := 0$$

$$bl5 := \frac{y_{[2,\,2]}}{y_{[1,\,1]}y_{[2,\,2]}-y_{[1,\,2]}y_{[2,\,1]}}+\frac{rl}{1+w^2\,cl^2\,rl^2}+rc \\ +\frac{rb\,cc\,(cc+cbc)}{rb^2\,cbc^2\,cc^2\,w^2+(cc+cbc)^2}+\frac{y_{[1,\,2]}}{y_{[1,\,1]}y_{[2,\,2]}-y_{[1,\,2]}y_{[2,\,1]}}-re$$

$$bl6 := w\,le$$

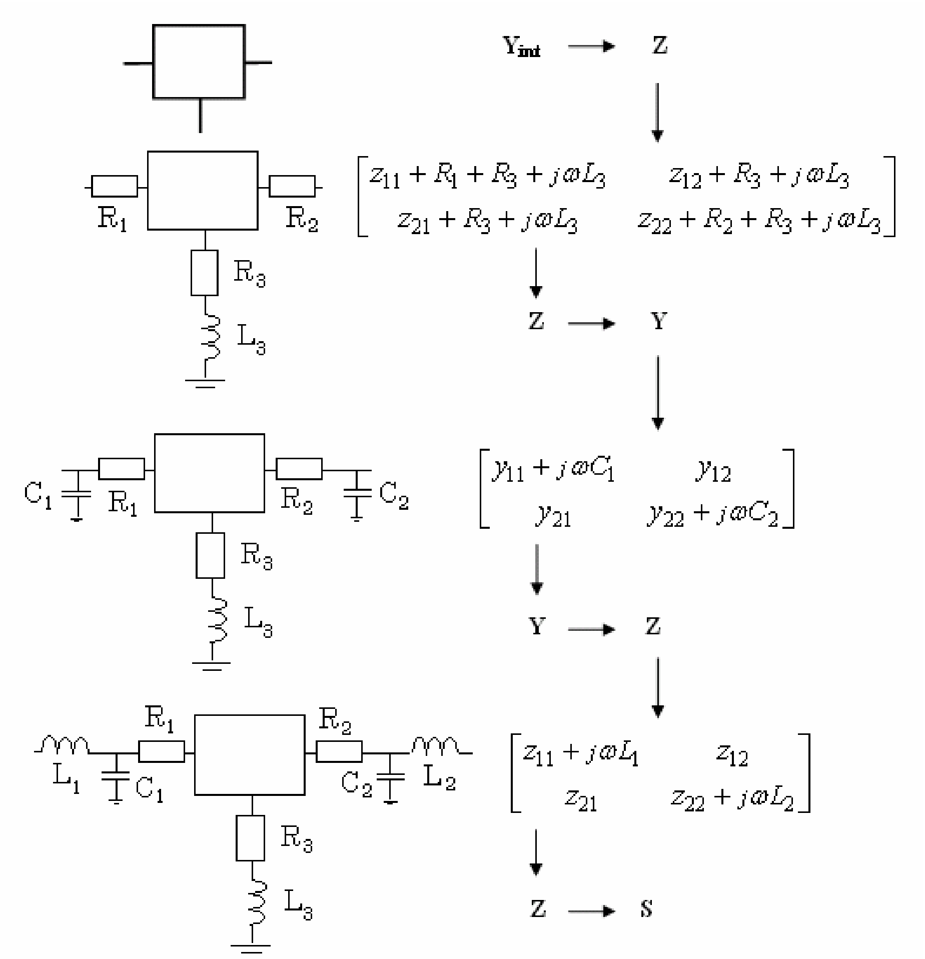
$\mathbf{Y_{int}} \rightarrow \mathbf{Z}$
R$_1$
R$_2$
R$_3$
L$_3$
$\begin{bmatrix} z_{11}+R_1+R_3+j\omega L_3 & z_{12}+R_3+j\omega L_3 \\ z_{21}+R_3+j\omega L_3 & z_{22}+R_2+R_3+j\omega L_3 \end{bmatrix}$
$\mathbf{Z} \rightarrow \mathbf{Y}$
C$_1$
R$_1$
R$_2$
C$_2$
R$_3$
L$_3$
$\begin{bmatrix} y_{11}+j\omega C_1 & y_{12} \\ y_{21} & y_{22}+j\omega C_2 \end{bmatrix}$
$\mathbf{Y} \rightarrow \mathbf{Z}$
R$_1$
R$_2$
L$_1$
C$_1$
C$_2$
L$_2$
R$_3$
L$_3$
$\begin{bmatrix} z_{11}+j\omega L_1 & z_{12} \\ z_{21} & z_{22}+j\omega L_2 \end{bmatrix}$
$\mathbf{Z} \rightarrow \mathbf{S}$

Fig. 14. Overall **S** matrix calculation from the intrinsic admittance matrix $\mathbf{Y_{int}}$.