Small-Signal Modeling for Microwave FET Linear Circuits Based on a Genetic Algorithm

Roberto Menozzi, *Associate Member, IEEE*, Aurelio Piazzi, *Member, IEEE*, and Fabrizio Contini

Abstract— **In this paper we present a new method to efficiently optimize small-signal equivalent circuits for microwave and millimeter-wave FET linear circuit design. The method couples the stochastic search of a Partially Elitistic Genetic Algorithm with a local search procedure. Up to 19 equivalent circuit elements have been included in the small-signal model for completeness and flexibility. Optimization examples are given for an ion-implanted MESFET up to 12 GHz, a pseudomorphic HEMT up to 50 GHz, and for synthetic data. The results show that the proposed algorithm is able to consistently provide an excellent fit between measured and calculated S-parameters without any need of a careful initial guess for the circuit element values. Also, once the device parasitics have been de-embedded, the algorithm is able to extract unique, physically meaningful values for the intrinsic device parameters, and it is numerically shown not to be affected by measurement uncertainties.**

I. Introduction

THE DESIGN OF MICROWAVE and millimeter-wave linear circuits such as low-noise amplifiers requires accurate modeling of the small-signal behavior of active components, typically GaAs MESFET's and GaAs- or InP-based MODFET's. Moreover, small-signal models (SSM's) are often used as a building block for large-signal modeling of nonlinear circuits (power amplifiers, oscillators, mixers) by extracting SSM parameters at different bias points and interpolating their bias dependence with empirical functions. Since a large number of equivalent circuit elements (ranging from 12 to 19 in most instances) is required if small-signal broadband behavior of microwave and millimeter-wave FET's has to be modeled, optimization of the whole set of SSM parameters by conventional, gradient-based techniques is a heavy computational task. Even more painful is the existence of several local minima in the error function (calculated as the difference between measured and modeled S-parameters over the frequency range of interest), which results in a marked sensitivity of the optimization results to the chosen initial trial, since it is well-known that gradient-based optimization algorithms tend to get stuck in local minima. Unfortunately, often only a very rough *a priori* knowledge of parameter values is available, resulting in serious difficulties in finding a proper starting point for the optimization process.

Manuscript received August 1, 1995; revised January 22, 1996. This paper was recommended by Associate Editor P. P. Civalleri.

R. Menozzi and A. Piazzi are with the Dipartimento di Ingegneria dell'Informazione, Università di Parma, Parma, Italy.

F. Contini was with the Dipartimento di Ingegneria dell'Informazione, Università di Parma, Parma, Italy. He is now with R. R. Robotica, Reggio Emilia, Italy.

Publisher Item Identifier S 1057-7122(96)07621-0.

To reduce the number of unknown model parameters, a common practice is to measure, as a first step of SSM extraction, the extrinsic (i.e., parasitic) elements, that are commonly considered to be bias-independent. This de-embedding of the extrinsic parameters is commonly performed as a first step, when conventional numerical optimization methods are used, to avoid nonuniqueness of results, i.e., different parameter sets that yield about the same device simulated response.

Also, a series of alternative extraction techniques has been proposed, among which direct methods, such as that of Dambrine *et al.*, [1] are the most widespread: here all of the circuit elements are extracted from suitably approximated analytical expressions using a few measurements in addition to that of the S-parameters at the bias point of interest. Such methods suffer from measurement inaccuracies, and along with others [2], [3] require that the intrinsic S-parameters be de-embedded from the measured ones by stripping off the contribution of the device parasitics, that have to be extracted by additional measurements. Some of these measurements must be performed under rather extreme bias conditions (e.g., with forward-biased gate) or require particular test structures on-chip, for instance to measure pad capacitances.

It would thus be most handy to come out with an optimization tool able to provide a good match between measured and modeled device frequency response, in reasonable times and with no need of careful initial estimates of the model parameters. Also, in the cases where the problem of nonuniqueness of results due to the large number of SSM parameters has been previously resolved by separately extracting the parasitics, the tool should be able to provide, again with no initial guess bias, a unique optimum solution, i.e., one whose elements may be considered to have a sound physical meaning.

Stochastic techniques such as Genetic Algorithms (GA's) and Simulated Annealing have been widely used for years in complex optimization problems; applications to microwave CAD can be found in [4]–[7]. In this paper we present for the first time a method for the optimization of small-signal equivalent circuits that makes use of a GA coupled with a semi-heuristic local exploration procedure. The method is particularly attractive because: 1) it is a global estimation method able to avoid the local minima problem of gradient-base techniques even when only a very rough estimate of the model parameter values is possible before optimization; 2) it is computationally efficient, meaning that complex SSM's can be satisfactorily optimized with computation times of a few minutes on low-cost PC platforms; 3) the stochastic search performed by our GA allows a large number of parameters (up

to 19 in the present work) to be included in the optimization process without eccessive computer overhead.

In the following section we describe the genetic algorithm; the optimization results are presented and discussed in Section III, and a few conclusions are summarized in Section IV.

II. The Genetic Algorithm

Starting with the pioneering work of Holland on artificial adaptation [8] many research contributions and applications of GA's have appeared in the literature: see for instance [9], [10], that also include extensive bibliography on this subject. In particular, the power and effectiveness of GA's in solving global optimization problems have emerged [11].

In this section we succintly expose the Partially Elitistic GA (PEGA) devised for FET small-signal model optimization. The outline of this algorithm is first given, followed by explanatory remarks and comments.

1) Generate randomly a population P of n individuals (strings of 0–1 b) and evaluate their *fitness*.
2) *Reproduction*: Create a *transient* population P_t, made up of n individuals, by copying, with a *fitness*-biased stochastic procedure, the individuals of P into P_t.
3) Apply genetic operators (*Crossover*, *Mutation*) to P_t.
4) Evaluate all of the individuals of P_t.
5) Extract randomly (without any *fitness* bias) $\eta \cdot n$ $(0 \leq \eta \leq 1)$ individuals of P and include them into P_t (*Partially Elitistic* procedure).
6) Choose deterministically the best n individuals out of the $n + \eta \cdot n$ of P_t to replace the population P.
7) Extract one individual randomly out of the m best ones, apply the semi-heuristic local exploration procedure to the extracted individual and put the modified individual (whose *fitness* has been improved or in the worst case has not changed) back in its place in the population P.
8) If the stopping criterion is not satisfied, go to 2.
9) Choose the best individual of P and exit.

A. The Population

Individuals (strings) of population P represent vector points of the model parameter space. Let us assume that the SSM features l parameters, whose values are indicated by $\pi_i (i = 1, \cdots, l)$. Once the user has defined the minimum and maximum value allowed for each parameter ($\pi_i^{\min}$ and $\pi_i^{\max}$, respectively), the value of every model parameter is coded with 12 b in the allowed range. Each string of $12 \cdot l$ b is built by merging the l 12-b parameter codings in such a way as to group together bits of the same importance (position in the 12-b parameter coding), starting with the most significant bits to end with the least significant ones. Put in another way, each string will begin with the l most significant bits of the parameter codings and end with the l least significant ones. This coding strategy makes the *Crossover* operator more effective. The program defaults at $n = 100$, i.e., P is made up of 100 individuals.

B. Fitness

The *fitness* is simply defined as $K - |\varepsilon|$, where $|\varepsilon|$ is the error norm defined in (1). K is a positive constant chosen large enough for the *fitness* to be always positive.

C. Reproduction

The *Reproduction* phase, at step 2, creates the transient population (P_t) out of the original one (P) by choosing, randomly but with a *fitness* bias, an individual from P, copying it in P_t, and repeating this procedure n times. Individuals with better *fitness* will be copied more often than low-*fitness* ones, and at the end of step 2 P_t will feature a better average *fitness* than P. Thus, this step implements a sort of *survival-of-the-fittest* phylosophy. *Reproduction* employs the Stochastic Remainder Selection Without Replacement [9], [12]. The necessary *fitness* scaling uses a linear scaling coefficient whose deafult value is 1.4 (see [9] for details).

D. Genetic Operators

Crossover and *Mutation*, in this order, are the genetic operators applied at step 3. First, all of the strings of P_t are randomly paired. Then, string pairs undergo *Crossover* with probability p_c (that defaults at 0.9). *Crossover* is an exchange of string fragments between the two strings of the pair, and is the most important tool by which the algorithm may generate and test new strings (i.e., parameter sets), thus exploring possible new solutions to the optimization problem. *Crossover* simulates the exchange of parent genes taking place in nature when a new offspring is born. In its simplest and by far most utilized form, *Crossover* between a pair of strings (individuals) is implemented by randomly selecting a crossing site within the string and swapping all bits that lie between the crossing site and the string end [9]. It follows that, if we choose a standard coding strategy where the string sections corresponding to the model parameters are simply concatenated to form the individual string of $12 \cdot l$ b, for each string pair *Crossover* will alter the value of a single parameter, namely the one containing the crossing site, while some parameters will be exchanged unaltered between the individuals of the pair. On the other hand, with the enhanced coding strategy described in point A and implemented in our algorithm, *Crossover* will in general alter all of the parameters, thus more effectively providing new parameter values to test. More importantly, with respect to standard coding, enhanced coding will make high-*fitness schemata* much shorter, hence less likely to be disrupted by *Crossover* [13]. Although the concept of *schema* (or *similarity template*), as many basic features of GA's, cannot be explained here for length reasons (the reader is once more invited to refer to [9] for a detailed exposition) a simple example may clarify this point. Suppose that a certain generation of the population P features a very high *fitness* individual, characterized, for instance, by high values of C_{GS}, R_{DS}, and a low value of G_M (see next section and Fig. 2 for the definition of the model parameters): this means that the highest order bit of the parameter string will be 1 for C_{GS} and R_{DS} and 0 for G_M. If enhanced coding is used, all highest order bits are grouped together, hence

the probability that the crossing site will fall among them is much lower than with standard coding, where the highest order bits are separated from one another by a whole parameter string length. In particular, with enhanced coding the three bits mentioned above (110), that constitute a *schema*, are close to one another, i.e., the *schema* is a short one, and *Crossover* is not likely to disrupt it and eliminate the high-*fitness* individual from the population. On the contrary, with standard coding the three bits will be far apart (a long *schema*), and there is a much higher probability that the crossing site will lie between two of them; if this happens, after *Crossover* the high-*fitness* individual may be lost.

Finally, the *Mutation* operator is applied bit-by-bit, with probability p_m (whose default value is $5 \cdot 10^{-4}$), to all of the individuals of P_t. *Mutation* simply makes a "1" become "0" and viceversa.

E. The Partially Elitistic Model

The proposed PEGA can be regarded as a special variant of both the Simple Genetic Algorithm (SGA) of Goldberg [9] and the Simple Elitistic Model (SEM) of De Jong [14]. Indeed, apart from the mechanism of local search inserted at step 7, our genetic algorithm exactly recovers the SGA if $\eta = 0$, whereas with $\eta = 1$ we get a variant of the purely elitistic SEM. The possibility of fixing an intermediate value of η, in the interval $(0, 1)$, has been proved to be very useful in our numerical experiments. With the default value of $\eta = 0.5$ the PEGA has turned out to be sharply superior to both SGA and SEM.

F. The Local Search

Before checking for *Genetic Saturation* (see below) and, if saturation is not reached, going back to step 2 and applying *Reproduction* to the new generation, a semi-heuristic local search takes place (similar to a coordinate descent method [15]) in the neighborhood of an individual extracted among the m fittest ones (m defaults at 10). This procedure explores a portion of the parameter space centered in the parameter vector representing the extracted individual; if a string with a better *fitness* is found, the extracted individual is replaced by the new one. With the default setting the algorithm explores about 2% of the allowed range of each parameter. This local search was seen to consistently improve the performance of the PEGA.

G. Genetic Saturation

The concept underlying the stopping criterion is *Genetic Saturation* in the overall population P: when the difference between the best *fitness* and the average *fitness* of P is smaller than a fixed threshold ε_{sat} the algorithm reaches the last generation and stops. The individual of P with the best *fitness* is taken as the solution of the optimization problem. Fig. 1 shows how *Genetic Saturation* was reached in one of the optimization runs described in the following section: both the error norm of the *fittest* individual of each generation and the average error of the generation are plotted versus the

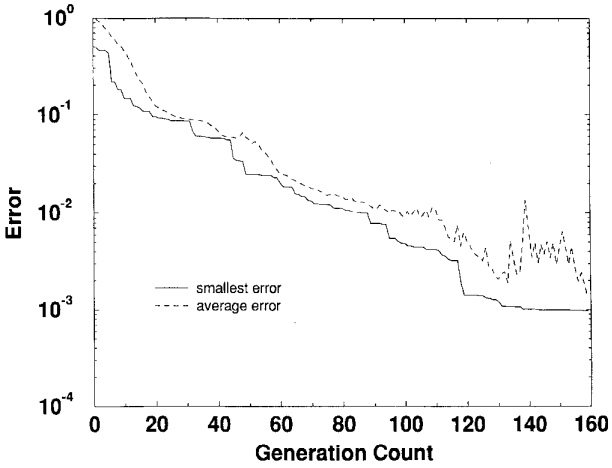


Fig. 1. Dependence of the error norm (calculated according to (1)) on the number of generations processed by the PEGA. *Genetic Saturation* is reached at generation 159, when the difference between the average error computed over the whole population (dashed line) and the error corresponding to the *fittest* individual (solid line) is smaller than $5 \cdot 10^{-4}$.

generation count. Here ε_{sat} is set to $5 \cdot 10^{-4}$, as in all of the experiments that will be described in the following section. Fig. 1 suggests a couple of considerations as to what value of ε_{sat} to use. First, ε_{sat} should be small enough to prevent premature saturation; for instance, choosing $\varepsilon_{\text{sat}} = 5 \cdot 10^{-3}$ in the case of Fig. 1 leads to saturation at generation 44, with an error about 60 times larger than the final one obtained with $\varepsilon_{\text{sat}} = 5 \cdot 10^{-4}$. On the other hand, if ε_{sat} is too small, saturation takes too long or cannot be reached altogether; Fig. 1 shows that, from generation 120 on, only marginal improvement of the smallest error is achieved, while stochastic noise on the average error unfruitfully delays saturation. A maximum number of generations is set to stop the program when saturation cannot be reached or the computation time becomes too long. In order to get the best tradeoff between the final error and the computation time, this maximum number of generations will have to vary from application to application.

Inputs to the program are two files containing the measured S-parameters, and the lower ($\pi_i^{\min}$) and upper ($\pi_i^{\max}$) bounds for each SSM parameter π_i, respectively, plus an optional file for the weights to be used in the calculation of the error norm. At run time, the user can modify the default values for the following PEGA parameters: 1) number of individuals in the population P (n), 2) linear scaling coefficient, 3) *Crossover* probability (p_c), 4) *Mutation* probability (p_m), 5) Partially Elitistic Model parameter (η), 6) number of high-*fitness* individuals among which the starting point for the local search is chosen (m), 7) portion of the parameter vector space explored by the local search, 8) *Genetic Saturation* threshold (ε_{sat}), 9) maximum number of generations. The defalut values have been chosen following the indications of the fundamental literature on GA's (e.g., [9]) as a starting point, then adjusting the parameter values with the goal of obtaining an algorithm able to provide low-error models in reasonable computation times, as described in the next section.

The PEGA described so far is implemented in C language.

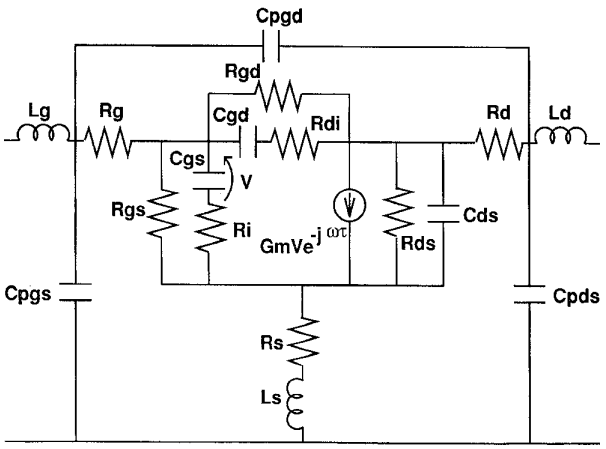


Fig. 2. Equivalent circuit topology for FET small-signal modeling.

III. Results

Fig. 2 shows the 19-parameter small-signal equivalent circuit topology that has been implemented in the program. Smaller sets of parameters can be used in most practical cases, as will be shown in the examples of this section. Nevertheless, broad-band modeling of millimeter-wave FET's requires that parasitic elements such as the pad capacitances (C_{PGS}, C_{PGD}, C_{PDS}) are taken into account; also, some high performance devices such as InP-based MODFET's often have rather leaky gates, the reverse currents of which must be modeled by the resistances R_{GS} and R_{GD}. Consequently, we have chosen the topology of Fig. 2 for its completeness and versatility. Whenever models with a reduced set of parameters can be satisfactorily used, the circuit can be simplified in a very straightforward fashion by simply assigning to the unnecessary elements suitable values of $\pi_i^{\min}$ and $\pi_i^{\max}$, so as to make their role in the circuit negligible (since the *fitness* that drives the convergence of the PEGA depends only on the difference between measured and modeled S-parameters, SSM's that have the same frequency response are equivalent to the PEGA, no matter if they comprise a different number of parameters).

As a first step, the *fitness* evaluation routine calculates the S-parameters of the SSM. This is done, at each frequency point, by first calculating the Y-parameters of the intrinsic device network, which has a suitable Π configuration, then converting the Y-matrix into the Z-matrix, adding the contributions of R_G, R_D, R_S and L_S, converting the Z-parameters back to Y-parameters, adding the contributions of C_{PGS}, C_{PGD} and C_{PDS}, going back one more time to a Z-matrix representation, including the contributions of L_G and L_D, and finally converting the Z-parameters into S-parameters. Having done this, the routine computes the error norm to be minimized, according to (1) shown at the bottom of the page, where $S_{ij}^m(f_k)$ and $S_{ij}^c(f_k)$ $(i,j = 1,2)$ are the measured and calculated S-parameters, respectively, at the frequency f_k, N_f is the number of frequency points, and $\max_{ij}$ is the maximum magnitude of S_{ij}^m. w_{ij} $(i,j = 1,2)$ are weighting coefficients that can be used when the required accuracy is not the same for all of the 4 S-parameters. The user may also decide, at run time, whether to use a first-order or a second-order error norm, by simply setting $q = 1$ or $q = 2$, respectively. $q = 1$ is often used in microwave device modeling due to its robustness in the presence of large errors [16]. $w_{ij} = 1$ $(i,j = 1,2)$ and $q = 1$ have been used in the simulations described in this section.

Three sets of optimization experiments have been performed and will be described hereafter. In the first one, the measured S-parameters of a 0.5 μm ion-implanted GaAs MESFET are modeled between 1 and 12 GHz using a 12-parameter SSM. This is done to evaluate the effectiveness of the PEGA as an optimizer and parameter extractor in a case where the frequency range is not very wide and the corresponding SSM is relatively simple. Although this case is far from representing state-of-the-art applications, it is of great practical relevance, since the vast majority of commercial microwave systems makes use of MESFET's operating at frequencies of a few GHz. The second experiment is aimed at testing the PEGA in a heavier task. A 0.25 μm $Al_{0.2}Ga_{0.8}As/In_{0.15}Ga_{0.85}As/GaAs$ pseudomorphic HEMT (PHEMT) is measured in the 2–50 GHz range, and its S-parameters are modeled using the complete 19-parameter SSM. Finally, in the third experiment, synthetic data have been used to test the ability of the PEGA to extract unique optimum solutions and to investigate the sensitivity of the extracted parameters to measurement errors. With the exception of the maximum number of generations allowed, which has to be changed from case to case to exploit the algorithm to its full potential, in all of the experiments we used the default values of the PEGA parameters.

The stochastic nature of GA's implies that different runs output different results. Nevertheless, a well-tuned GA able to solve its optimization problem will yield results (i.e., optimized error norm values) with limited spread, as will be shown below for our PEGA. Thus, in order to thoroughly and reliably characterize the performance of the PEGA, we present for each of the experiments the results of 10 optimization runs.

All of the optimization experiments have been performed on a 66 MHz 486 PC.

A. MESFET SSM Optimization

The S-parameters of an ion-implanted GaAs MESFET fabricated by GEC-Marconi (UK) were measurend on-wafer from 1 to 12 GHz with frequency steps of 1 GHz. The device features a 0.5 μm gate length and a 150 μm periphery, resulting in a unity current gain cut-off frequency $f_T = 20$ GHz. The bias point is $V_{GS} = 0.3$ V, $V_{DS} = 3$ V.

The MESFET behavior was modeled using a 12-parameter SSM, because: i) due to via-hole grounding, L_S was found to

$$|\varepsilon| = \frac{1}{4N_f} \times \left(\sum_{i,j=1}^{2} \sum_{k=1}^{N_f} \frac{w_{ij}\left(\left|\mathrm{Re}\left(S_{ij}^m(f_k) - S_{ij}^c(f_k)\right)\right|^q + \left|\mathrm{Im}\left(S_{ij}^m(f_k) - S_{ij}^c(f_k)\right)\right|^q\right)}{\max_{ij}^q} \right)^{1/q} \tag{1}$$

TABLE I
SUMMARY OF THE RESULTS OF THE MESFET SSM OPTIMIZATION EXPERIMENTS

Run	Final error	Generations processed	Computation time
#1	0.117	119	4'55"
#2	0.0142	180	7'34"
#3	0.0120	181	7'30"
#4	0.0109	100	4'9"
#5	0.0298	81	3'30"
#6	0.0111	152	6'21"
#7	0.0239	147	6'14"
#8	0.0116	168	6'53"
#9	0.0232	200	8'16"
#10	0.0841	98	4'7"

TABLE II
MINIMUM, MAXIMUM, AND OPTIMUM VALUE FOR EACH PARAMETER OF THE MODEL USED FOR THE MESFET SSM OPTIMIZATION

Parameter	π_i^{min}	π_i^{max}	π_i^{opt}
L_G $[nH]$	0.05	5	0.160
R_G $[\Omega]$	0.2	20	0.572
R_S $[\Omega]$	0.2	20	0.645
L_D $[nH]$	0.05	5	0.0597
R_D $[\Omega]$	0.2	20	0.200
C_{GS} $[fF]$	50	5000	191
R_I $[\Omega]$	0.2	20	6.95
C_{GD} $[fF]$	5	500	14.5
G_M $[mS]$	5	500	24.8
τ $[ps]$	0.5	50	3.11
R_{DS} $[\Omega]$	50	5000	436
C_{DS} $[fF]$	5	500	64.8

be negligible; ii) the pad capacitances (C_{PGS}, C_{PGD}, C_{PDS}) were also negligible, as can be expected for on-wafer measurements in the 1–12 GHz frequency range; iii) the gate leakage was low enough for R_{GS} and R_{GD} to be disregarded; iv) R_{DI} was found not to be necessary for a good match between model and measurements; the effect of R_{DI} becomes appreciable at lower V_{DS} (close to the linear region) and higher frequencies. Since, as often happens, the model elements could be estimated only roughly *a priori*, we choose a worst-case condition where the parameter values span over two orders of magnitude, i.e., $\pi_i^{\max}/\pi_i^{\min} = 100$ $(i = 1, ..., 12)$.

The final error values, the number of generations processed and the computation times required are reported for each of the 10 runs in Table I.

The final error is quite low in most of the cases in Table I, which is an indication that the proposed PEGA is an effective tool to find a satisfactory solution to the optimization problem. Fig. 3 shows measured (dots) and calculated (lines) S-parameters for the 5 best runs among the 10 of Table I. The match with the measured data is excellent.

We have thus shown that our PEGA is able to satisfactorily optimize the 12-parameter SSM to match the measured

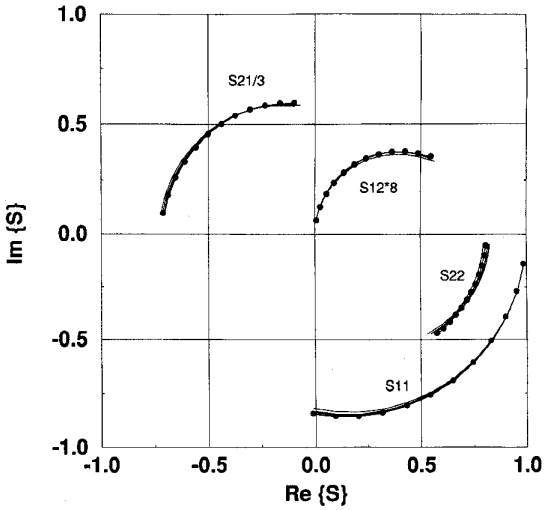


Fig. 3. Measured (dots) and calculated (lines) MESFET S-parameters in the 1–12 GHz frequency range. The calculated data refer to the 5 best runs of Table I.

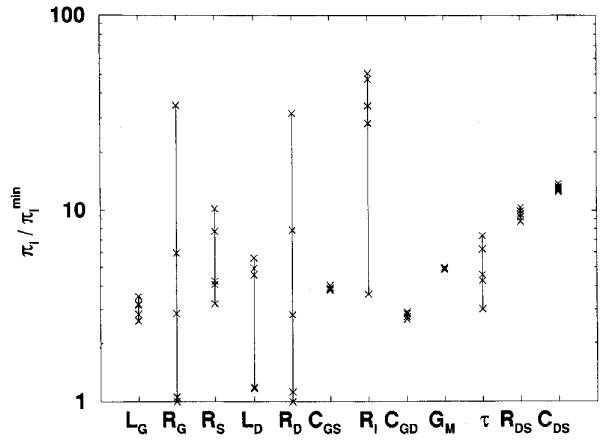


Fig. 4. SSM parameter values extracted by the 5 best runs of Table I. Each parameter is normalized to the lower bound of its allowed range.

MESFET S-parameters. As far as the extraction of the model parameters is concerned, however, the algorithm has to face the well-known problem of nonuniqueness of solutions, meaning that, when the number of circuit elements is high, different parameter sets may yield very similar calculated equivalent circuit responses. The situation is illustrated in Fig. 4, where we reported normalized values of the extracted parameters corresponding to the data of Fig. 3, i.e., to the 5 best runs among the 10 of Table I. Since they are connected in series with the larger impedances of the intrinsic elements, the parasitics (L_G, R_G, R_S, L_D, R_D) cannot be extracted accurately; the problem with R_I (in series with C_{GS}) is similar. However, it has to be noticed that, with the exception of R_I (and, to a lesser extent, of τ), the intrinsic elements can be extracted with very good accuracy. Table II reports $\pi_i^{\min}$, $\pi_i^{\max}$ and the optimum value (i.e., the one that corresponds to the lowest-error run of the PEGA), π_i^{opt}, for each model parameter.

The nonuniqueness of solutions is a concern when a physical description of the device is sought with the SSM.

TABLE III
SUMMARY OF THE RESULTS OF THE PHEMT SSM OPTIMIZATION EXPERIMENTS

Run	Final error	Generations processed	Computation time
#1	0.0416	94	5'49"
#2	0.0428	187	10'25"
#3	0.0644	252	14'21"
#4	0.0248	319	18'18"
#5	0.0270	297	16'51"
#6	0.0272	279	15'47"
#7	0.0622	109	6'30"
#8	0.0322	266	16'7"
#9	0.0350	184	11'17"
#10	0.0334	290	17'47"

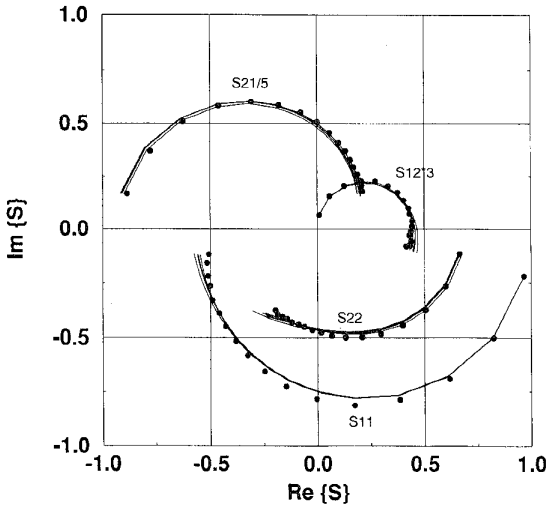


Fig. 5. Measured (dots) and calculated (lines) PHEMT S-parameters in the 2–50 GHz frequency range. The calculated data refer to the 5 best runs of Table III.

In such instances, parasitic elements must be extracted before the optimization process by means of additional measurements and/or suitable test structures. To test the ability of the PEGA to extract unique values for the intrinsic equivalent circuit elements once the parasitic shell has been de-embedded, we have performed optimization experiments on synthetic data, as described in Section III-C.

B. PHEMT SSM Optimization

The S-parameters of δ-doped $Al_{0.2}Ga_{0.8}As/In_{0.15}$-$Ga_{0.85}$-As-GaAs PHEMT's manufactured at IMEC (Belgium) were measured on-wafer between 2 and 50 GHz with 3 GHz steps. The devices feature <0.25 μm gate length, 100 μm gate width and $f_T \simeq 80$ GHz. The bias point was $V_{GS} = 0.3$ V, $V_{DS} = 2$ V.

Since the frequency range is much wider than in the MESFET experiment, we adopted the complete 19-parameter SSM to fit the measured S-parameters. This represents, to the authors' knowledge, the most complete SSM topology used for microwave and millimeter-wave FET's, and a huge computational task for standard, gradient-based optimizers: it is therefore common practice to reduce the number of unknowns by performing additional measurements (e.g., to extract the parasitics) before feeding the data to the optimizer. The stochastic nature and the efficiency of our PEGA allow to bypass this problem: as we show below, very good fit between measured and calculated data can consistently be obtained by a direct optimization of the whole 19-parameter set in reasonable computation times. As in the MESFET case, to evaluate the PEGA performance under worst-case conditions, we assume that very limited *a priori* knowledge of the parameter values is available, and assume $\pi_i^{\max}/\pi_i^{\min} = 100$ $(i = 1, \cdots, 19)$.

TABLE IV
MINIMUM, MAXIMUM, AND OPTIMUM VALUE FOR EACH PARAMETER OF THE MODEL USED FOR THE PHEMT SSM OPTIMIZATION

Parameter	π_i^{min}	π_i^{max}	π_i^{opt}
L_G $[nH]$	0.005	0.5	0.0479
R_G $[\Omega]$	0.5	50	7.97
L_S $[nH]$	0.005	0.5	0.0110
R_S $[\Omega]$	0.5	50	2.68
L_D $[nH]$	0.005	0.5	0.0257
R_D $[\Omega]$	0.5	50	4.14
C_{PGS} $[fF]$	1	100	24.5
C_{PDS} $[fF]$	1	100	18.2
C_{PGD} $[fF]$	1	100	3.01
R_{GS} $[M\Omega]$	5	500	394
C_{GS} $[fF]$	10	1000	86.4
R_I $[\Omega]$	0.5	50	1.55
R_{GD} $[M\Omega]$	5	500	485
C_{GD} $[fF]$	1	100	18.8
R_{DI} $[\Omega]$	0.5	50	10.4
G_M $[mS]$	10	1000	65.8
τ $[ps]$	0.001	0.1	0.0977
R_{DS} $[\Omega]$	10	1000	227
C_{DS} $[fF]$	1	100	30.8

Table III summarizes the results obtained, and measured data are compared with the 5 best runs of the PEGA in Fig. 5. Given the wide frequency range adopted, the fit provided by the algorithm can be considered a very good one, and the spread among the different runs is quite limited. As could be expected, the average number of generations processed to get to genetic saturation is larger than in the previous MESFET experiment, since the optimization problem is now much more complex. As far as the computation times are concerned, however, once we normalize the times in Tables I and III to the number of frequency points, the average CPU time for the PHEMT optimization is about 45% longer than its MESFET counterpart, which is a very moderate increase, considered that we are now concurrently optimizing 19 parameters instead of 12.

Finally, $\pi_i^{\min}$, $\pi_i^{\max}$, and π_i^{opt}, for each model parameter, are shown in Table IV. As pointed out above in the discussion of the MESFET experiment, the high number of model parameters precludes the possibility of extracting a unique

TABLE V
SUMMARY OF THE RESULTS OF THE SYNTHETIC DATA EXPERIMENTS

Run	Final error	Generations processed	Computation time
#1	0.0077	114	8'38"
#2	0.0135	86	6'45"
#3	0.0053	84	6'33"
#4	0.0058	77	6'7"
#5	0.0016	116	9'12"
#6	0.0053	111	8'36"
#7	0.00097	159	12'10"
#8	0.0126	78	6'1"
#9	0.0079	115	9'11"
#10	0.0080	128	9'57"

TABLE VI
MINIMUM, MAXIMUM, OPTIMUM, AND "TRUE" VALUE FOR EACH PARAMETER OF THE MODEL USED FOR THE SYNTHETIC DATA EXPERIMENTS

Parameter	π_i^{min}	π_i^{max}	π_i^{opt}	π_i^t
C_{GS} $[fF]$	10	1000	180	180
R_I $[\Omega]$	0.1	10	2.65	2.50
C_{GD} $[fF]$	1	100	25	25
G_M $[mS]$	10	1000	62	62
τ $[ps]$	0.1	10	0.83	0.85
R_{DS} $[\Omega]$	10	1000	190	189
C_{DS} $[fF]$	1	100	48.7	49.0

set of parameter values by means of a direct optimization of the complete model, no matter which optimization tool we use. For instance, since the parasitic pad capacitor C_{PGS} is practically in parallel with the intrinsic C_{GS}, the optimizer will be able to extract the value of $C_{PGS} + C_{GS}$, but not to separate the contribution of the individual capacitances. Consequently, whenever one needs to attribute a precise physical meaning to the model parameters, there is no other option but the de-embedding of the device parasitic shell by means of additional measurements or *ad hoc* test structures.

C. Synthetic Data Experiments

We have numerically generated synthetic data using a set of values for the equivalent circuit elements that corresponds to a FET with $f_T = 47$ GHz. The S-parameters, calculated between 0.5 and 23.5 GHz with 1 GHz steps, represent our "measured" device frequency response. The SSM used here is the one that represents the intrinsic device only and features 7 parameters (we keep $R_{GS} = R_{GD} = \infty$ and $R_{DI} = 0$), namely: C_{GS}, R_I, C_{GD}, G_M, τ, R_{DS}, C_{DS}. Since in this case the ability of the PEGA to extract unique circuit element values had to be tested, once more we let the SSM parameters span over a very wide range by choosing $\pi_i^{\max}/\pi_i^{\min} = 100$ $(i = 1, \cdots, 7)$.

As in the previous experiments, 10 optimization runs were performed, as summarized by Table V.

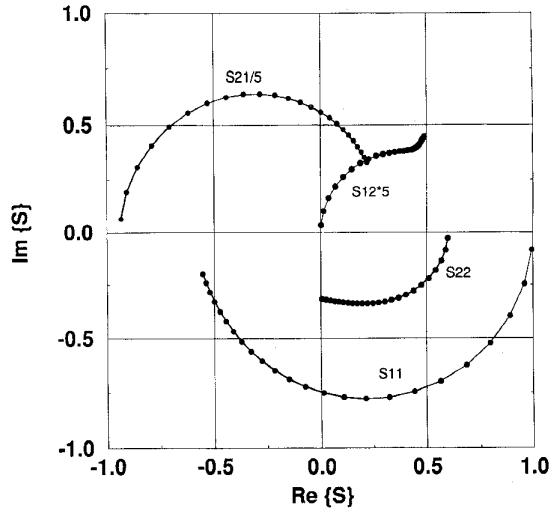


Fig. 6. "Measured" (dots) and calculated (lines) S-parameters for the synthetic data experiment. The calculated data refer to the 5 best runs of Table V. The frequency ranges from 0.5 to 23.5 GHz.

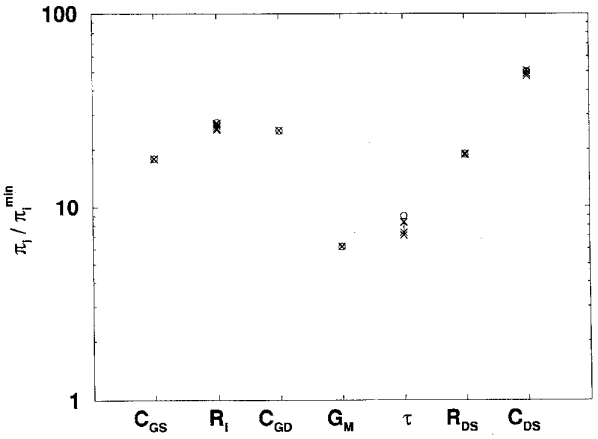


Fig. 7. SSM parameter values extracted by the 5 best runs of Table V. Each parameter is normalized to the lower bound of its allowed range. The element values used to generate the synthetic data are shown by circles.

The final errors are extremely low in this case, ranging from less than 0.1 to 1.3%. Computation times are somewhat longer than in the MESFET experiment due to the larger number of frequency points that directly affects the CPU time dedicated to *fitness* evaluation. Measured (dots) and calculated (lines) S-parameters for the 5 best runs among the 10 of Table V (practically sitting on top of each other) are plotted in Fig. 6, indicating that excellent fit is consistently achieved by the PEGA.

Fig. 7 shows the normalized parameter values extracted by the best 5 runs of Table V, together with "true" values, i.e., those used to generate the synthetic data (circles). The spread among the results is negligible, with extracted values tightly gathered in the vicinity of "true" values. $\pi_i^{\min}$, $\pi_i^{\max}$, and π_i^{opt} are shown, for each model parameter, together with the "true" values (π_i^t), in Table VI.

It follows from the above discussion that, once the problem of nonunique solutions is got rid of by previous de-embedding

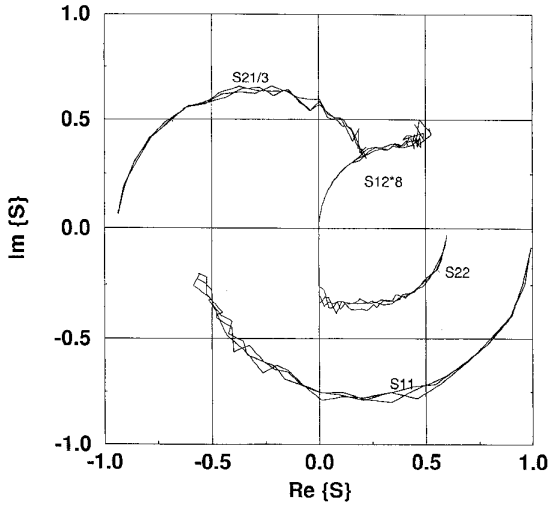


Fig. 8. Synthetic data generated to simulate three different S-parameter measurements in the presence of measurement errors. The frequency ranges from 0.5 to 23.5 GHz.

of the device parasitics, the PEGA is able to provide both an excellent fit of the measured data and a unique set of parameters for the SSM, thus allowing a sound physical meaning to be associated to the equivalent circuit elements.

Another important issue is the sensitivity of the extracted model parameters to the measurement errors. In particular, this point is a critical one when direct extraction techniques such as the one presented in [1] are used. To investigate the sensitivity of the results provided by our optimization algorithm with respect to measurement inaccuracies, we have simulated three sets of S-parameter measurements including the effect of experimental errors. The errors were calculated by generating three different sequencies of random numbers with the same gaussian distribution function; each sequency corresponds to a single measurement of the four S-parameters in the 0.5–23.5 GHz frequency range. Since for each of the four S-parameters we have to account for the error on both the real and the imaginary part, and since our frequency scan features 24 points, each sequence of random numbers is made of $4 \times 2 \times 24 = 192$ elements. To account for the increase of the error magnitude with measurement frequency, the random numbers were multiplied by a linearly increasing function of the frequency. Finally, in order to consider a reasonable worst-case condition that can be encountered with up-to-date equipment, the straggle of the gaussian distribution was chosen in such a way that about 90% of the random numbers fall between -0.1 and 0.1. The resulting "measured" S-parameters are shown, for the three experiments, in Fig. 8. The results of the application of our PEGA to the error-affected data of Fig. 8 are summarized in Tables VII and VIII. In particular, Table VII shows the final errors encountered in ten optimization runs for each one of the three experiments, and demonstrates that the spread of the error values from run to run is quite negligible, i.e., the accuracy of the fit is not limited by the optimization algorithm but by the spread of the measured data. The average values of computation times and of the number of generations processed are practically

TABLE VII
FINAL ERRORS OF THE THREE EXPERIMENTS WITH ERROR-AFFECTED DATA

Run	Exp. #1	Exp. #2	Exp. #3
#1	0.0289	0.0267	0.0258
#2	0.0267	0.0262	0.0272
#3	0.0263	0.0255	0.0299
#4	0.0238	0.0260	0.0260
#5	0.0220	0.0261	0.0263
#6	0.0233	0.0276	0.0270
#7	0.0224	0.0258	0.0293
#8	0.0245	0.0256	0.0261
#9	0.0232	0.0259	0.0255
#10	0.0250	0.0257	0.0272

TABLE VIII
PARAMETER VALUES EXTRACTED BY THE PEGA IN THE THREE EXPERIMENTS WITH ERROR-AFFECTED DATA, TOGETHER WITH THE "TRUE" VALUES π_i^t

Parameter	Exp. #1	Exp. #2	Exp. #3	π_i^t
C_{GS} $[fF]$	180	180	180	180
R_I $[\Omega]$	2.16	2.51	2.32	2.50
C_{GD} $[fF]$	25.1	24.9	24.8	25
G_M $[mS]$	62	62	62	62
τ $[ps]$	0.96	0.89	0.77	0.85
R_{DS} $[\Omega]$	188	188	190	189
C_{DS} $[fF]$	48.6	49.4	50.5	49.0

identical to those corresponding to the error-free experiment of Table V.

What is most important though is the ability of the optimization technique to extract unique, i.e., physically meaningful parameter values even in the presence of measurement errors. The excellent performance of the PEGA is illustrated in Table VIII, where the values of the model parameters extracted by the PEGA (the lowest-error case has been chosen for each experiment) are reported for each one of the three experiments of Fig. 8 and Table VII, together with the parameter values used to generate the synthetic data (the "true" values, π_i^t). As in the previously described experiments, the parameter values were allowed to span over two orders of magnitude. Table VIII indicates that even in the presence of substantial measurement errors the PEGA is able to precisely extract unique values of the model parameters.

IV. CONCLUSION

We have presented a method for the optimization of small-signal equivalent circuits of microwave and millimeter-wave FET's based on a genetic algorithm coupled with a semi-heuristic local search procedure. The algorithm has been successfully applied to both measured and synthetic data in a wide frequency range, and we showed that it is able to: i) consistently provide excellent model fitting and ii)

once the nonuniqueness of solutions has been overcome by preliminary de-embedding of the device parasitics, extract unique, physically meaningful values for the equivalent circuit elements even in the presence of significant experimental errors.

The computation times required depend on several factors, first of all the number of frequency points to be processed, but generally range between 4 and 15 minutes on a 66 MHz 486 PC. Obviously enough, shorter times will be needed using faster PC platforms or workstations.

One of the most interesting features of the proposed genetic algorithm is that it requires practically no *a priori* knowledge of the equivalent circuit parameter values. In the optimization experiments described in the paper we have shown that even though these values are allowed to span over two orders of magnitude, convergence to a very good fit can consistently be achieved. This means that the initial-guess bias of conventional nonlinear programming techniques is not an issue for our algorithm, which is an important result of this paper.

With straightforward modifications, namely a change of the routine that calculates the model S-parameters, the algorithm can be used to optimize small-signal equivalent circuits of different topologies, such as those used for bipolar transistors; in principle, it may also be applied to a wide range of large-signal modeling problems, whenever an efficient and initial-guess insensitive optimization tool is required.

ACKNOWLEDGMENT

The authors are grateful to G. Green of GEC-Marconi, UK, for providing the MESFET's, to Y. Baeyens of IMEC, Belgium, for the PHEMT's, and to G. Vannini of the University of Bologna, A. Santarelli of the University of Bologna, and M. Borgarino of the University of Parma, for the S-parameter measurements.

REFERENCES

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

[2] F. Lin and G. Kompa, "FET model parameter extraction based on optimization with multiplane data-fitting and bidirectional search—A new concept," *IEEE Trans. Microwave Theory Tech.*, vol. 42, pp. 1114–1121, July 1994.

[3] L. T. Wurtz, "GaAsFET and HEMT small-signal parameter extraction from measured S-parameters," *IEEE Trans. Instrum. Meas.*, vol. 43, pp. 655–658, Aug. 1994.

[4] G. L. Bilbro, M. B. Steer, R. J. Trew, C.-R. Chang, and S. G. Skaggs, "Extraction of the parameters of equivalent circuits of microwave transistors using tree annealing," *IEEE Trans. Microwave Theory Tech.*, vol. 38, pp. 1711–1718, Nov. 1990.

[5] M.-K. Vai, J.-S. Lin, and S. Prasad, "Acceleration of simulated annealing and its application to microwave device and circuit optimization," in *IEEE MTT-S Int. Microwave Symp. Dig.*, June 1991, pp. 1213–1216.

[6] S. G. Skaggs, J. Gerber, G. Bilbro, and M. B. Steer, "Parameter extraction of microwave transistors using a hybrid gradient descent and tree annealing approach," *IEEE Trans. Microwave Theory Tech.*, vol. 41, pp. 726–729, Apr. 1993.

[7] E. Michielssen, J.-M. Sajer, S. Ranjithan, and R. Mittra, "Design of lightweight, broad-band microwave absorbers using genetic algorithms," *IEEE Trans. Microwave Theory Tech.*, vol. 41, pp. 1024–1031, June/July 1993.

[8] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor MI: The Michigan Univ. Press, 1975.

[9] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. New York: Addison-Wesley, 1989.

[10] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin: Springer-Verlag, 1992.

[11] Z. Michalewicz, G. A. Vignaux, and L. Groves, "Genetic algorithms for optimization problems," in *Proc. 11th New Zeland Computer Conf.*, Wellington, New Zeland, 1989, pp. 211–223.

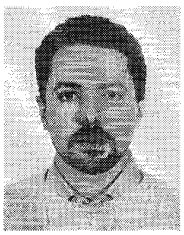
[12] L. B. Booker, "Intelligent behavior as an adaptation to the task environment," Ph.D. dissertation, Univ. Michigan, *Dissertation Abstracts International*, vol. 43, no. 2, p. 469B, 1982.

[13] T. Kozek, T. Roska, and L. O. Chua, "Genetic algorithm for CNN template learning," *IEEE Trans. Circuits Syst. I*, vol. 40, pp. 392–402, June 1993.

[14] K. A. De Jong, "An analysis of the behavior of a class of genetic adaptive systems," Ph.D. dissertation, Univ. Michigan, *Dissertation Abstracts International*, vol. 36, no. 10, p. 5140B, 1975.

[15] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*, 2nd ed. New York: Addison-Wesley, 1984, p. 227.

[16] J. W. Bandler, S. H. Chen, and S. Daijavad, "Microwave device modeling using efficient l_1 optimization: A novel approach," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-34, pp. 1282–1293, Dec. 1986.



Roberto Menozzi (S'91–A'95) was born in Genova, Italy, in 1963. He received the Laurea degree in electronic engineering from the University of Bologna in 1987 and the Ph.D. degree in information technology from the University of Parma in 1994.

After serving in the army, he joined a research group at the Department of Electronics, University of Bologna. Since 1990, he has been with Dipartimento di Ingegneria dell'Informazione, University of Parma. Since 1993 he has been a Research Associate with the Dipartimento di Ingegneria dell'Informazione, University of Parma. His research activity has dealt with the study of latch-up in CMOS circuits, IC testing, and DC, RF and noise characterization, modeling and reliability evaluation of compound semiconductor devices such as GaAs MESFET's and GaAs- and InP-based HEMT's.

Aurelio Piazzi (M'92) received the Laurea degree in nuclear engineering in 1982 and the Ph.D. degree in system engineering in 1987, both from the University of Bologna.

From 1990 to 1992 he was Reasearch Associate in System Theory, D.E.I.S., University of Bologna. Since November 1992 he has been Associate Professor of Automatic Control, Dipartimento di Ingegneria dell'Informazione, University of Parma. His main research interests are in system and control theory and related engineering applications. His current research activity concerns the methods of global optimization applied to the analysis and design of robust control systems.

Dr. Piazzi is a member IFAC and SIAM.

Fabrizio Contini was born in Reggio Emilia, Italy, in 1968. He received the Laurea degree in electronic engineering from the University of Parma in 1995. His final year's project dealt with the implementation of a genetic algorithm for FET small signal model optimization.

Since his graduation, he has been with R. R. Robotica, Reggio Emilia, where he is involved in software development for automation and robotics.