Microelectronics Reliability 49 (2009) 660–666

Contents lists available at ScienceDirect

Microelectronics Reliability

journal homepage: www.elsevier.com/locate/microrel

MESFET DC model parameter extraction using Quantum Particle Swarm Optimization

Samrat L. Sabat[a,*], Leandro dos Santos Coelho[b], Ajith Abraham[c]

[a] School of Physics, University of Hyderabad, Hyderabad 500 046, India
[b] Pontifica Catholic University of Parana, PUCPR Industrial and Systems Engineering Graduate Program, PPGEPS, Zip Code 80215-901 Curitiba, PR, Brazil
[c] Q2S, Norwegian Center of Excellence, NTNU, Trondheim, Norway

ARTICLE INFO

Article history:
Received 8 December 2008
Received in revised form 2 February 2009
Available online 22 April 2009

ABSTRACT

This paper presents two techniques for DC model parameter extraction for a Gallium Arsenide (GaAs) based MEtal Semiconductor Field Effect Transistor (MESFET) device. The proposed methods uses Particle Swarm Optimization (PSO) and Quantum Particle Swarm Optimization (QPSO) methods for optimizing the difference between measured data and simulated data. Simulated data are obtained by using four different popular DC models. These techniques avoid complex computational steps involved in traditional parameter extraction techniques. The performance comparison in terms of quality of solution and execution time of classical PSO and QPSO to extract the model parameters are presented. The validity of this approach is verified by comparing the simulated and measured results of a fabricated GaAs MESFET device with gate length of 0.7 μm and gate width of 600 μm (4 × 150). Simulation results indicate that both the technique based on PSO and QPSO accurately extracts the model parameters of MESFET.

1. Introduction

Device simulation is one of the important steps for Integrated Circuit fabrication, verification and characterization. Each semiconductor device have models that satisfies the behavior of the device under different operating conditions. GaAs MESFET is a promising semiconductor device used in many applications in the microwave domain. Many models of GaAs MESFET device are reported in the literature [1–5]. Each model has a unique set of parameters that describes the underlying physical phenomena of the device. These parameters are obtained by minimizing the difference between measured drain current and modeled drain current at different gate bias voltage. This process is known as parameter extraction. Usually the model parameters are extracted using commercial software like HP IC-CAP [6], Silvaco UTMOST [7], TMA AURORA [8], etc. Parameter extraction is an crucial and difficult step for the circuit and device simulator. Until recent past, the model parameter extraction were carried out by using standard gradient based algorithm or Levenberg Marquardt (LM) algorithm [9]. LM algorithm is sensitive to initial values of parameters and is thus proned to be trapped in local minima. The complexity of the model lead the extraction algorithm to take longer computation time for resulting the solution. The traditional algorithm is not an ideal approach if more parameters need to be extracted from a complex model at a time. The major drawbacks of this approach is its poor convergence rate and non-optimal solutions. Non-optimal solutions are due to trap of algorithmic solutions in local optima. To overcome these drawbacks, genetic algorithm (GA) is being used in parameter extraction of semiconductor device model [10]. The advantages of using GA is that its solutions are independent on initial values of the parameters and it provides an optimal set of solution by avoiding local non-optimal solutions. The major drawbacks of GA based approach are that it involves more algorithmic steps and it provides inconsistent results in different simulation environment.

To overcome this problem, we have proposed two techniques namely PSO and QPSO algorithm for model parameter extraction. The PSO algorithm was first introduced by Kennedy and Eberhart in 1995 [11]. Many variants of PSO algorithm were developed by the authors [12–14] to improve the quality of solution. In recent past, PSO is being used for solving complex optimization problems [15–17]. The authors have also applied PSO algorithm for extracting small signal model parameters of MESFET [18]. The popularity of this algorithm is due to its simple form, easy implementation steps and ability to avoid local minima. The basic PSO still suffers the poor convergence rate. In this paper, a new variant of PSO known as delta potential well quantum PSO (DQPSO) based on quantum mechanics is used to extract different DC model parameters of a fabricated MESFET device. QPSO algorithm has proven to have advantages than the classical PSO due its less control parameters [19,20]. In recent past, DQPSO is used to solve real world complex problems [21,22]. More details about QPSO algorithm is presented in

[*] Corresponding author. Tel.: +91 40 23134326.
E-mail addresses: slssp@uohyd.ernet.in (S.L. Sabat), leandro.coelho@pucpr.br (L. dos Santos Coelho), ajith.abraham@ieee.org (A. Abraham).

0026-2714/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.microrel.2009.03.005

Section 5. The performance comparison of both classical PSO and QPSO algorithm for model parameter extraction is presented. In this paper, all the simulation are carried out for a single geometry device structure. The methodology proposed in this paper is to optimize the error between measured and simulated data using PSO and DQPSO algorithms. The simulated data were obtained using different popular device models [2–5].

The rest of paper is organized as follows: Section 2 provides a brief description of the model parameter extraction strategies. Section 3 presents the problem formulation for parameter extraction and a brief description about classical PSO algorithm is provided in Section 4. Section 5 presents the description about Quantum Particle Swarm Optimization followed by simulation and results in Section 6. Conclusions are drawn in Section 7.

2. Model parameter extraction

The accuracy of a commercial available software used for device simulation depends on the accuracy of the device models and the parameter extraction algorithm being used in the software. A model is accurate if it fits the measured data in all the operating regions of the characteristics curve. In order to closely resemble between model data and measured data, more numbers of parameters are included in the model to describe the behavior of device accurately. The inclusion of more number of parameters in the model results in a complex model and eventually parameter extraction algorithm becomes more complex. As the number of parameters in a model increases, the conventional parameter extraction algorithm fails to provide accurate values. So an efficient parameter extraction algorithm is always in demand for fabrication industry. The efficiency of the algorithm depends on the quality of solutions and convergence time.

Two strategies are often used in parameter extraction domain. One is local optimization in which each parameter is extracted from certain region of the characteristics curve. This can be interpreted as, local optimization strategy is region specific. The disadvantages of this technique are the mismatch between simulated and measured data and the designer must have domain knowledge to use this strategy. The second strategy is to adopt a global optimization technique. In this technique, the wide set of parameters are extracted from a wide range of measured data and is not region specific. The designer only needs to give a range of values of the variables to be determined. This input helps the algorithm to avoid non-physical solutions. In this paper, a new variant of PSO based on quantum mechanics is used to extract the model parameters of the MESFET model.

3. Formulation of parameter extraction problem

The main objective of DC model parameter extraction problem is to minimize the difference between measured and simulated drain current I_d at various drain source V_{ds} and gate source V_{gs} voltage of MESFET. It can be formulated using an objective function. In this paper, the objective function is the square of the difference between the measured drain current and the simulated drain current. The choice of objective function affects the numerical efficiency of the algorithm. In this paper, for obtaining the simulated data, four different MESFET DC models are considered such as TOM3 model, Tajima Model, Materka–Kacprzak Model and Curtice–Ettenburg model [1]. Mathematically the objective function can be represented as

$$E = \sum_{V_{gs}=1}^{P} E_{V_{gs}} \tag{1}$$

$$E_{V_{gs}} = \sum_{V_{ds}=1}^{K} (I_{sim,V_{ds},V_{gs}} - I_{meas,V_{ds},V_{gs}})^2 \tag{2}$$

where P is the number of gate bias voltage (V_{gs}) and $I_{sim,V_{ds},V_{gs}}$ is simulated drain current at drain bias voltage V_{ds} and gate bias voltage V_{gs}. $I_{meas,V_{ds},V_{gs}}$ is the corresponding measured drain current value at drain bias voltage V_{ds} and gate bias voltage V_{gs}. Let us consider the simulated model as Tajima model [2]. In case of Tajima model the drain current is given by

$$I_{ds}(V_{ds}, V_{gs}) = I_{d1}.I_{d2} \tag{3}$$

where

$$I_{d1} = \frac{1}{k}\left[1 + \frac{V'_{gs}}{V_p} - \frac{1}{m} + \frac{1}{m} exp\left\{-m\left(1 + \frac{V'_{g}s}{V_p}\right)\right\}\right] \tag{4}$$

$$I_{d2} = I_{dsp}\left[1 - exp\left\{\frac{-V_{ds}}{V_{dsss}} - a\left(\frac{V_{d}S}{V_{dss}}\right)^2 - b\left(\frac{V_{d}S}{V_{dss}}\right)^3\right\}\right] \tag{5}$$

where $k = 1 - \frac{1}{m}\{1 - exp(-m)\}, V'_{gs} = V_{gs} - V_{\phi}$ and $V_p = V_{p0} + pV_{ds} + V_{\phi}$. Where V_{ϕ} is the built in potential for the schottky barrier, V_{p0} is the pinch-off voltage at $V_{ds} \sim 0$ V. *a,b,m* and *p* are the fitting parameters. V_{dss} is the drain current saturation voltage and I_{dsp} is the drain current at $V_{p0} = V_{\phi}$. The parameter extraction problem is formulated as

$$minimize(f) = \sum_{V_{gs}=1}^{P} E_{V_{g}s} \tag{6}$$

From Eq. (2) the objective function can be expressed as:

$$f = \left[\sum_{V_{gs}} \sum_{V_{ds}} \frac{(I_{meas,V_{ds},V_{gs}} - I_{sim,V_{ds},V_{gs}})^2}{I_{meas,V_{ds},V_{gs}}}\right]^{1/2} \tag{7}$$

Objective function is a function that needs to be optimized for extracting different parameters. If the number of parameters to be extracted in the model increases, minimizing Eq. (7) using traditional optimization technique becomes computational intensive. A simple PSO and QPSO algorithms are proposed in this paper and used to minimize Eq. (7) for extracting the model parameters of different MESFET models. Careful selection of the proper range of parameters to be optimized are considered to avoid the danger of blind optimization. The model parameters need to be optimized are model dependent. The algorithmic parameters of PSO and QPSO are considered same for the considered device models [2–5].

4. Classical Particle Swarm Optimization Algorithm

PSO algorithm is a stochastic and robust optimization algorithm based on intelligence and movements of birds in the Swarm [11]. It has been applied to many real world optimization problems successfully [15–18]. In the PSO domain, each bird is termed as a single particle. Each particles position are potential solution of the optimization problem. The number of variables to be optimized decides the number of dimensions of the optimization problem. Each particle is associated with two numbers of variables known as position and velocity. To find the optimal solution, a Swarm of Particles are initialized with a random position and velocity within a defined range. Positions are initialized within a range defined by X_{max} and X_{min}. Velocity is initialized between 0 to V_{max}, where V_{max} is $\{X_{max} - X_{min}\}/2$. Each particle adjusts its "flying trajectory" according to its own best position (*pbest*) as well as the swarm's best position (*gbest*). Each particle's strength in the search process is evaluated based on the fitness value of the particle. While in the search process each particle updates their *pbest* in the following way. Each particle remembers its previous solution, and if the current solution is better than the previous solution, the position corresponding to current solution becomes the *pbest*, otherwise the position corresponds to the previous solution is the *pbest*. The best among the *pbest* solution is the *gbest* solution and the position corresponds to *gbest* value are the potential solution of the problem. This process is known as updating *pbest* and *gbest.*

The practical implementation of classical PSO involves the following steps.

1. Initialization of algorithmic parameters: In this step, initialization of the number of dimension along with the range, number of particles along with position and velocity in the Swarm is defined. For a D-dimensional problem with N number of particles the position vector is represented as $X(t) = (X_1(t), X_2(t), X_3(t), \ldots, X_N(t))$ where $X_i = (x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iD})$ and the velocity vector is represented as $V(t) = (V_1(t), V_2(t), V_3(t), \ldots, V_N(t))$, where $V_i = (v_{i1}, v_{i2}, v_{i3}, \ldots, v_{iD})$.
2. Evaluate the desired objective function of each particle: In this step, objective function for each particle is evaluated.
3. Update *pbest*: In this step, each particles current fitness value is compared with previous best value *pbest*. If the current value is better than the *pbest* value, then set the *pbest* value to the current value.
4. Update *gbest*: In this step, determine the swarm best *gbest* as minimum of all the particles *pbest*.
5. Update the velocity and position of each particle: In this step velocity and position of each particle is updated as

$$v_{i,d}^{t+1} = v_{i,d}^{t} + c_1^t * rand_1 * (pbest_{i,d}^t - x_{i,d}^t) + c_2^t * rand_2 * (gbest_d^t - x_{i,d}^t) \quad (8)$$

$$x_{i,d}^{t+1} = x_{i,d}^{t} + v_{i,d}^{t+1} \quad (9)$$

   where c_1 and c_2 are the learning factors that determines the relative influence of cognitive and social component, respectively, in search space. In this paper, the values of c_1 and c_2 were decreased (adapted) with each iteration [11]. $rand_1$ and $rand_2$ are random numbers within the range [0,1]. $v_{i,d}^t, x_{i,d}^t$ and $pbest_{i,d}^t$ are the velocity, position and the personal best of ith particle in dth dimension for the tth iteration respectively. The $gbest_d^t$ is the dth dimension of best particle in the swarm for the tth iteration.
6. Terminate: In this step, the algorithm checks for stopping criteria, if it satisfies the criteria, then the algorithm terminates.

5. Quantum Particle Swarm Optimization

In the classical PSO, the particle moves in the search space by following Newtonian dynamics. Although classical PSO converges to the global solution, still for some problems it is not a global optimization technique, since it gets trapped in local minima. Classical PSO has many control parameters. The convergence of the algorithm depends on the value of the control parameters. Tuning a proper value for convergence of PSO algorithm is a tedious work. To avoid this problem a new PSO, which has only one control parameter and in which the movement of particles are inspired by the quantum mechanics [21,22] is proposed for extracting the model parameters of MESFET. Since the particles behavior follows the quantum mechanics principle, the PSO is termed as Quantum PSO (QPSO). In quantum mechanics the state of each particle is determined by a wave function $\psi_{j,k}(t)$ and is associated with the time dependent Schrödingers equation

$$j\hbar \frac{\partial}{\partial t}\psi(r,t) = \widehat{H}(r)\psi(r,t) \quad (10)$$

where $\widehat{H}$ is the time-independent Hamiltonian operator of the quantum system given by

$$\widehat{H}(r,t) = -\frac{\hbar^2}{2m}\nabla^2 + V(r) \quad (11)$$

where $\hbar$ is the Planck's constant and m is mass of the particle.*V(r)* is the potential energy of the distribution. The solution of Eq. (10) $\psi(r,t)$ gives the state of the particle in the system.

The choice of potential well plays an important role in QPSO. The necessary condition for choosing the potential well is the potential field of the potential well should give bound states for the particles moving in quantum states. Most popular potential well that satisfies this condition are delta potential well, harmonic oscillator potential well and square potential well. If the particles are in delta potential well then the corresponding QPSO algorithm is termed as delta well quantum PSO (DQPSO). Similarly if the particles are in harmonic oscillator well and square potential well the corresponding PSO are termed as harmonic oscillator QPSO (HQPSO) and square QPSO (SQPSO), respectively. In this paper, we used the delta well quantum PSO to extract DC model parameters of MESFET.

5.1. Delta Well Quantum Particle Swarm Optimization (DQPSO)

In DQPSO each particle move in the search space within a delta potential on each dimension with center at $p_{i,j}$. Index i and j are for particle i of jth dimension. The probability density function Q of particle $x_{i,j}$ can be expressed as [22]

$$Q(x_{i,j}(t+1)) = \frac{1}{L_{i,j}(t)} exp - \frac{|p_{i,j}(t) - x_{i,j}(t+1)|}{L_{i,j}(t)} \quad (12)$$

where $L_{i,j}(t)$ is characteristic length of potential well. By solving Eq. (12) using Monte-carlo technique the position of particle is obtained as

$$x_{i,j}(t+1) = p_{i,j}(t) \pm \frac{L_{i,j}(t)}{2} ln\frac{1}{u} \quad (13)$$

where u is a random number in the range [0,1]. Using the idea of center of mass position, the mean best position of all the particles can be defined as

$$m(t) = (m1(t), m2(t), \ldots, m_D(t)) = \left(\frac{1}{N}\sum_{i=1}^{N} p_{i,1}(t), \frac{1}{N}\sum_{i=1}^{N} p_{i,2}(t), \ldots, \frac{1}{N}\sum_{i=1}^{N} p_{i,D}(t)\right) \quad (14)$$

where N is the population size. $pbest_{i,j}$ is the best position of ith particle in jth dimension. The value of $L_{i,j}(t)$ can be determined by

$$L_{i,j}(t) = 2\alpha|x_{i,j}(t) - m_j(t)| \quad (15)$$

where α is contraction–expansion constant. Eq. (13) can be expressed as

$$x_{i,j}(t+1) = p_{i,j}(t) \pm \alpha \cdot |x_{i,j}(t) - m_j(t)| ln\frac{1}{u} \quad (16)$$

where m_j is the mean best of all the particles in jth dimension. This equation is implemented using Monte-carlo technique as

$$x_{i,j}(t+1) = p_{i,j}(t) + \alpha \cdot |x_{i,j}(t) - m_j(t)| ln\frac{1}{u} \quad \text{if } k > 0.5 \quad (17)$$

$$x_{i,j}(t+1) = p_{i,j}(t) - \alpha \cdot |x_{i,j}(t) - m_j(t)| ln\frac{1}{u} \quad \text{if } k < 0.5 \quad (18)$$

where k is a random number in the range [0,1]. In this paper, α is linearly decreasing factor from 1.0 to 0.3 with iteration as

$$\alpha^t = \alpha_{max} - \frac{\alpha_{max} - \alpha_{min}}{t_{max}} \cdot t \quad (19)$$

where t_{max} is the maximum number iteration used in algorithm. The practical implementation of DQPSO involves the following steps.

1. Initialization of algorithmic parameters: In this step initialization of the number of dimension along with the range, number of particles along with position in the swarm is defined. For a D-dimensional problem with N number of particles the position vector is represented as $X(t) = (X_1(t), X_2(t), X_3(t), \ldots, X_N(t))$ where $X_i = (x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iD})$.
2. Evaluate m and α, of Swarm using Eqs. (14) and (19), respectively.
3. Evaluate the desired objective function of each particle: In this step, the objective function for each particle is evaluated.
4. Update *pbest*: Each particles current fitness value is compared with previous best value *pbest*. If the current value is better than the *pbest* value, then set the *pbest* value to the current value.
5. Update *gbest*: Determine the swarm best *gbest* as minimum of all the particles *pbest*.
6. Evaluate the coordinate of local attractor for each particle in each dimension

$$p_{i,j}(t) = \frac{rand_1 \cdot pbest_{i,j}(t) + rand_2 \cdot gbest_j(t)}{rand_1 + rand_2} \tag{20}$$

7. Update the position of particle as Eqs. (17) and (18).
8. Repeat steps 2–7 until the stop criteria is satisfied.
9. Terminate: The algorithm checks for stopping criteria, if it satisfies the criteria, then the algorithm terminates.

6. Experimental results and analysis

All experiments were conducted in a Windows XP Professional, OS environment using a Pentium IV, 2.0 GHz, 2GB RAM and the codes were implemented in Matlab.

6.1. Parameter settings

Different Models of same dimension MESFET of gate width $W = 600\ \mu m$ (4×150) and gate length $L = 0.7\ \mu m$ are simulated for 50 times using the proposed techniques. Simulations are carried out with a population size (N) of 20, the number of iterations (Max.Gen) 1000. The dimensions of the problem is model dependent. The algorithm is tested on different MESFET models and compared with measured data from the specified MESFET. The model parameters of different models [2–5] for 0.7 μm technology GaAs MESFET are extracted using PSO and DQPSO technique. Algorithmic parameters of PSO and DQPSO are tabulated in Table 1. The values of the extracted parameters for the four different models using classical PSO and Quantum PSO are presented in Tables 2 and 3. Each model has different set of model parameters that are extracted from the measured drain current values. The mean square error (MSE) between the measured and simulated drain current has been calculated for each model using PSO and DQPSO technique and are presented in Table 4. In both the PSO and DQPSO algorithms, the set of parameters need to be extracted are initialized with random values within a predefined range. Predefined range of values are used to avoid non-physical solutions. The extracted values for different models using PSO and DQPSO for 50 different runs are presented in Tables 2 and 3. Although both methods provided good solutions, the computation time per iteration in PSO algorithm is much smaller compared to DQPSO method. The advantages of using DQPSO is that it has only one tuning parameter. The computation time per iteration varies from model to model and is tabulated in Table 4. In Table 4, T_{PSO} and T_{DQPSO} defines the

Table 1
Algorithmic parameters of PSO and DQPSO.

Algorithmic parameter	PSO	DQPSO
No. of particle	20	20
Max. iteration	1000	1000
c_1, c_2	Varies with iteration	–
α	–	Varies with iteration

Table 2
List of extracted parameters using PSO algorithm.

Model name	Parameters	*Min*	*Max*	Mean	Std deviation
Materka [2]	I_{DSS}	0.1	0.1	0.1	0.00
	V_{t0}	−1.93	−1.83	−1.89	0.02
	α	3.16	5.43	4.02	0.77
	γ	−0.14	−0.12	−0.13	0.01
Tajima [3]	V_{BI}	0.03	2.77	0.41	0.65
	V_{P0}	−2.61	1.99	1.60	1.08
	P	0.0	0.18	0.09	0.03
	m	−2.98	1.47	0.46	0.95
	I_{dsp}	0.01	0.19	0.12	0.03
	V_{DSS}	0.85	1.49	1.08	0.16
	a	0.08	3.89	2.89	0.83
	b	−0.28	0.86	0.41	0.29
TOM3 [4]	α	2.09	3.22	2.57	0.31
	β	0.02	0.04	0.03	0.01
	λ	0.0	0.01	0.0	0.00
	γ	0.01	0.02	0.01	0.00
	V_{to}	−2.42	−1.91	−2.13	0.14
	M_{sto}	0.08	0.19	0.14	0.03
	Q	1.53	2.01	1.75	0.14
	V_{sto}	0.03	0.07	0.05	0.01
CE [5]	β	0.03	0.07	0.04	0.01
	A_0	0.1	0.1	0.1	0.00
	A_1	0.07	0.11	0.09	0.01
	A_2	−0.04	0.08	0.03	0.02
	A_3	−0.05	0.04	0.01	0.02
	γ	1.43	2.76	1.99	0.32
	V_{ds0}	0.12	4.34	2.18	1.18

Table 3
List of extracted parameters using DQPSO algorithm.

Model name	Parameters	*Min*	*Max*	Mean	Std deviation
Materka [2]	I_{DSS}	0.1	0.1	0.1	0.00
	V_{t0}	−1.94	−0.37	−1.81	0.3
	α	3.12	5.84	4.26	0.64
	γ	−0.53	−0.11	−0.15	0.08
Tajima [3]	V_{BI}	0.01	0.97	0.31	0.27
	V_{P0}	1.71	2.0	1.89	0.08
	P	0.06	0.14	0.1	0.02
	m	−0.72	1.36	0.47	0.54
	I_{dsp}	0.1	0.2	0.13	0.03
	V_{DSS}	0.8	1.55	1.07	0.22
	a	1.23	4.0	2.91	0.8
	b	−0.6	0.97	0.31	0.42
TOM3 [4]	α	2.04	3.49	2.8	0.3
	β	0.01	0.04	0.02	0.01
	λ	0.0	0.06	0.01	0.01
	γ	0.01	0.04	0.01	0.01
	V_{to}	−2.45	−1.87	−2.17	0.14
	M_{sto}	0.09	0.2	0.14	0.03
	Q	1.53	2.43	1.82	0.18
	V_{sto}	0.03	0.08	0.05	0.01
CE [5]	β	−0.01	0.05	0.03	0.02
	A_0	0.1	0.12	0.1	0.00
	A_1	0.05	0.17	0.09	0.02
	A_2	−0.04	0.14	0.02	0.04
	A_3	−0.05	0.04	0.01	0.02
	γ	1.49	2.86	2.02	0.29
	V_{ds0}	0.33	4.84	2.24	1.12

Table 4
Performance comparison between PSO and DQPSO.

Model name	PSO MSE	DQPSO MSE	T_{PSO} (s)	T_{DQPSO} (s)	No. of parameters
Materka [2]	0.000031	0.00091	0.0069	19.49	4
Tajima [3]	0.0018	0.0013	0.0155	48.94	8
TOM3 [4]	0.246	0.0122	0.0055	37.3219	8
Curtice and Ettenberg (CE) [5]	0.0003	0.122	0.0043	28.1609	7

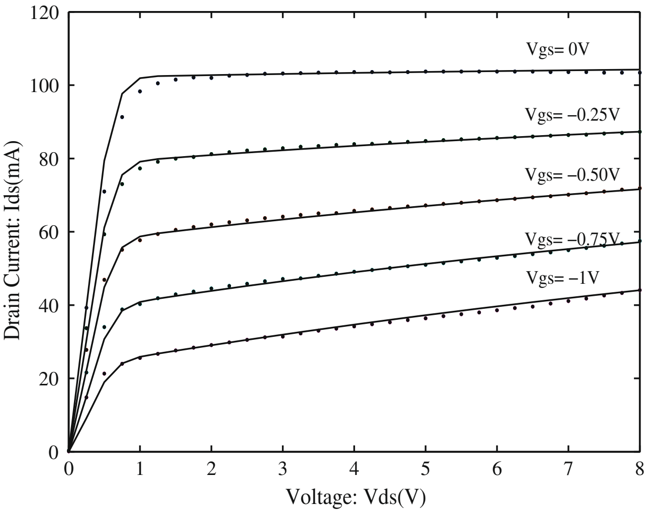

Fig. 1. Simulated (—) and measured (···) DC characteristics of MESFET for Tajima model.

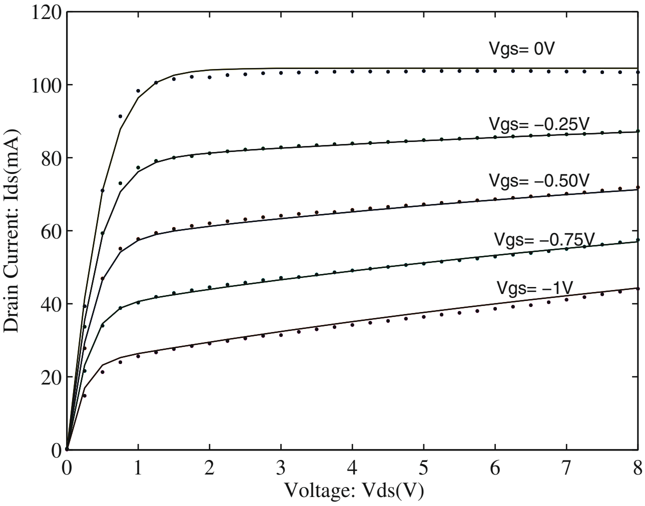

Fig. 3. Simulated (—) and measured (···) DC characteristics of MESFET for Materka model.

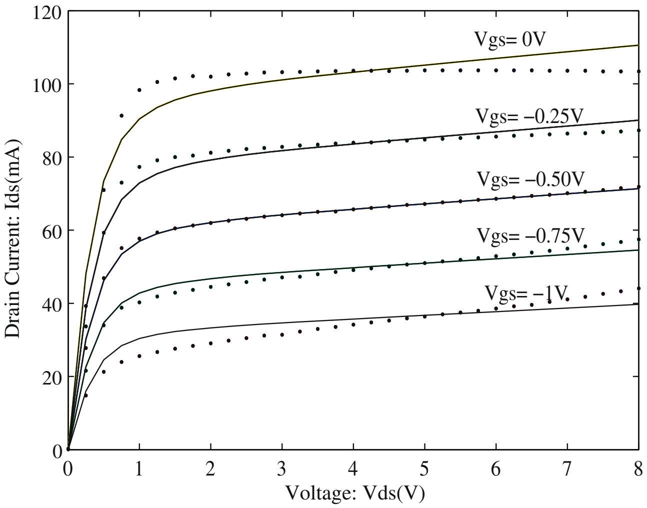

Fig. 2. Simulated (—) and measured (···) DC characteristics of MESFET for TOM3 model.

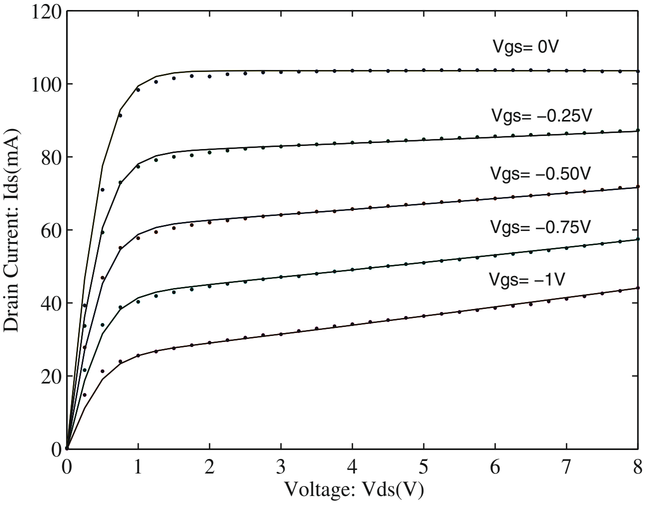

Fig. 4. Simulated (—) and measured (···) DC characteristics of MESFET for Curtice and Ettenberg model.

computation time per iteration using PSO and DQPSO technique, respectively. Figs. 1–4 illustrate the output characteristics of the fabricated MESFET. The solid line represents the simulated data and the doted line represents the measured data using proposed methods. Fig. 5 shows the characteristics of $Log(Ids)$ vs. V_{gs} of the MESFET for TOM3 model. For $V_{gs} \ll V_{threshold}$ the model fails to model the drain current accurately. Figs. 6–8 shows the derivative of drain current with respect to different lead voltages using proposed parameter extraction technique and TOM3 model. From Figs. 1–8, it is clear that the proposed parameter extraction algorithm is able to accurately extracts the model parameters of the device. The PSO and DQPSO provides almost same results and the PSO approach gives less mean square error compared to DQPSO. The performance illustration is shown only for PSO. A comprehensive verification is carried out on different popular MESFET models [1–5].

For PSO algorithm c_1 and c_2 varies with iteration t as follows:

$$c_1^t = c_{1max} + \frac{c_{1max} - c_{1min}}{Max.iter} \times (Max.iter - t) \qquad (21)$$

and similarly

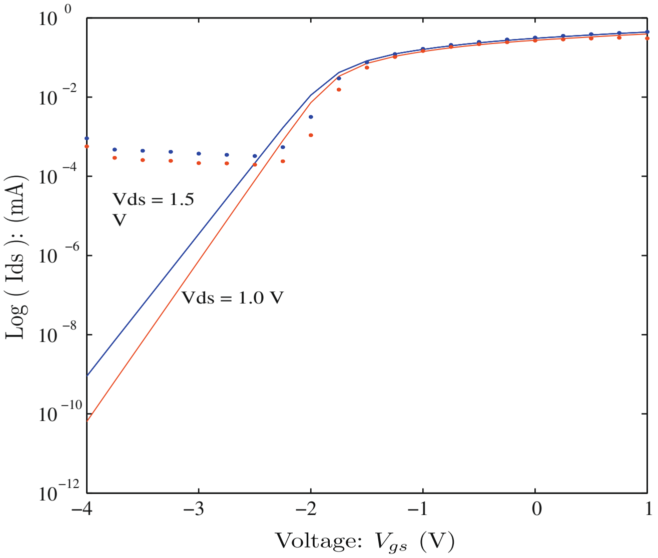

Fig. 5. Simulated (—) and measured (···) Log(Ids) vs. V_{gs} of MESFET for TOM3 model.

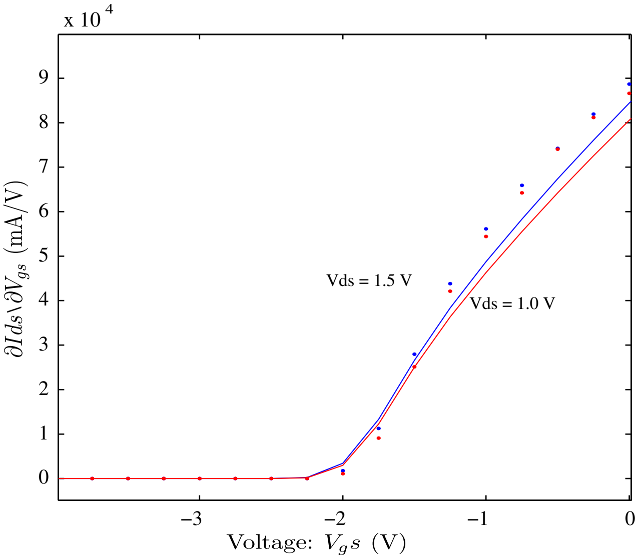

Fig. 7. Simulated (—) and measured (···) $\partial(Ids)/\partial(V_{gs})$ vs. V_{gs} of MESFET for TOM3 model.

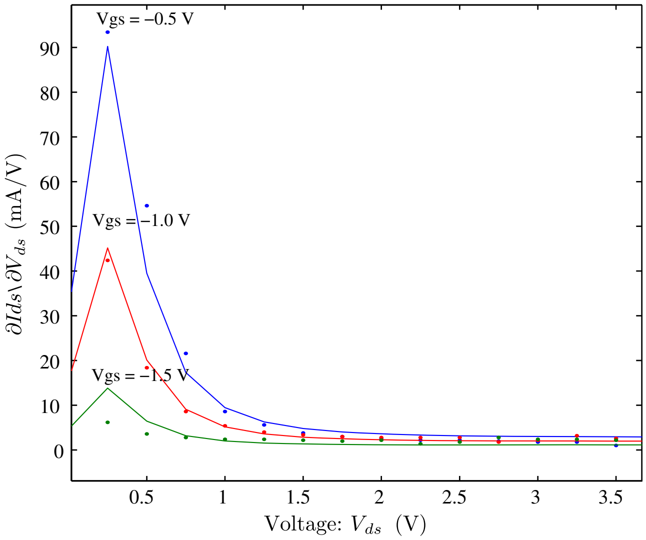

Fig. 6. Simulated (—) and measured (···) $\partial(Ids)/\partial(V_{ds})$ vs. V_{ds} of MESFET for TOM3 model.

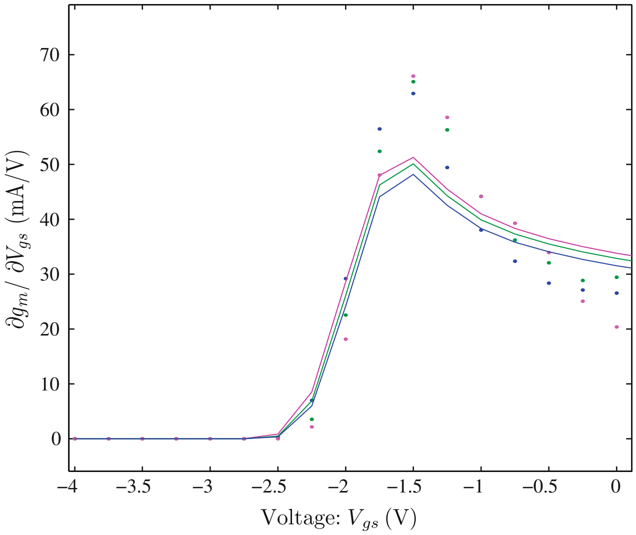

Fig. 8. Simulated (—) and measured (···) $\partial(g_m)/\partial(V_{gs})$ vs. V_{gs} at different V_{ds} bias points 1 V (red), 1.5 V (green) and 2 V (blue) using TOM3 model. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$c_2^t = c_{2max} + \frac{c_{2max} - c_{2\,min}}{Max.iter} \times (Max.iter - t) \quad (22)$$

where *Max.iter* is the maximum number of iteration used in simulation. c_{1max} and c_{2max} are same as 0.9 and c_{1min} and c_{2min} are same as 0.2.

7. Conclusion

In his paper, the extraction of model parameters of four different models using PSO and DQPSO has been investigated. The performance comparison of both the algorithms are carried out for different models. Empirical results indicate that PSO and DQPSO algorithms are efficient optimization techniques for model parameter extraction. However DQPSO technique required more computational time compared to the classical PSO algorithm. The proposed technique does not require any expert knowledge in MESFET device and are not sensitive to initial value of the parameters for extracting the model parameter values. The accuracy of the developed technique is verified in terms of the DC characteristic simulation and measured data of a GaAs MESFET device. Materka model and TOM3 model are the suitable model for the fabricated MESFET. The proposed method can be interfaced to commercial tools such as IC-CAP for extracting the different parameters of semiconductor device.

Acknowledgement

The authors are thankful to the Department of Science and Technology and DRDO, India for providing necessary support to carry out the research work.

References

[1] Memon NM, Ahmed MM, Rehman F. A comprehensive four parameters IV model for GaAs MESFET output characteristics. Solid-State Elect 2007;51:511–6.
[2] Materka A, Kacprzak T. Computer calculation of large-signal GaAs FET amplifier characteristics. IEEE Trans Microw Theory 1985;33:129–35.
[3] Tajima Y, Wrona B, Mishima K. GaAs FET large-signal model and its application to circuit designs. IEEE Trans Electron Dev 1981;28:171–5.
[4] Hallgen Robert B, Litzenberg Paul H. TOM3 capacitance model: linking large- and small-signal MESFET models in SPICE. IEEE Trans Microw Theory 1999;47:556–61.
[5] Curtice WR, Ettenberg M. A nonlinear GaAs FET model for use in the design of output circuits for power amplifiers. IEEE Trans Microw Theory 1985;33:1383–93.
[6] Hewlett-Packard Co. IC-CAP user's manual; 1994.
[7] Silvaco International. UTMOST III modeling manual. Santa Clara, CA; 1992.
[8] Technolong Modeling Associates, Inc. AURORA-device characterization system; 1994.
[9] Marquardt D. An algorithm for least squares estimation of nonlinear parameters. J Soc Ind Appl Math 1963;11:431–41.
[10] Gao Yifan. Genetic algorithm based parameter extraction for power GaAs MESFET. Microw Rev 2001;12:47–8.
[11] Kennedy J, Eberhart RC. Particle swarm optimization. In: IEEE proceedings of the conference on neural networks, vol. 4; 1995. p. 1942–8.
[12] Sabat Samrat L, Ali Layak. The hyperspherical acceleration effect particle swarm optimizer, Appl Soft Comp J. <http://dx.doi.org/10.1016/j.asoc.2008.11.003>.
[13] Pant M, Thangaraj R, Grosan C, Abraham A. Improved particle swarm optimization with low-discrepancy sequences. In: Proceedings of the IEEE congress on evolutionary computation; 2008. p. 3011–8.
[14] Pant M, Thangaraj R, Abraham A. Particle swarm optimization using adaptive mutation. In: IEEE proceedings of the 19th international conference on database and expert systems application; 2008. p. 519–23.
[15] Bhattacharya A, Konar A, Das S, Grosan C, Abraham A. Hardware software partitioning problem in embedded system design using particle swarm optimization Algorithm. In: IEEE proceedings of the international conference on complex, intelligent and software intensive systems; 2008. p. 171–6.
[16] Pant M, Thangaraj R, Abraham A. Optimization of a kraft pulping system using particle swarm optimization and differential evolution. In: IEEE proceedings of the second asia international conference on modeling and simulation; 2008. p. 637–41.
[17] Shuyuan Yang, Min Wang, Licheng jiao. A quantum particle swarm optimization. In: Proceedings of the IEEE congress on evolutionary computation, vol. 1; 2004. p. 320–4.
[18] Sabat Samrat L, Vijayaraju M, Ali Layak. MESFET small signal model parameter extraction using particle swarm optimization. In: Proceedings of the 20th IEEE international conference on microelectronics; 2008. p. 183–6.
[19] Robinson J, Samii Yahya Rahmath. Particle swarm optimization in electromagnetic. IEEE Trans Antenna Prop 2004;52:397–400.
[20] Mikki SM, Kishk AA. Quantum particle swarm optimization for electromagnetics. IEEE Trans Antenna Prop 2006;54:2764–75.
[21] Sun J, Feng B, Xu Wenob. particle swarm optimization with particles having quantum behavior. In: Proceedings of the IEEE congress on evolutionary computation; 2004. p. 291–4.
[22] Sun Jun, Liu Jing, Xu Wenbo. Using quantum-behaved particle swarm optimization algorithm to solve non-linear programming problems. Int J Comp Math 2007;84(2):261–72.