Parameter Identification of a Power Amplifier Behavioral Model Using a Derivative-free Damping Switchable Levenberg-Marquardt Method

Felipe P. Ribeiro
Group of Integrated Circuits and Systems (GICS)
Federal University of Parana
Curitiba, Brazil
felipe.pires@ufpr.br

Eduardo G. Lima
Group of Integrated Circuits and Systems (GICS)
Federal University of Parana
Curitiba, Brazil
elima@eletrica.ufpr.br

***Abstract*—The behavioral modeling of a power amplifier (PA) is an indispensable step in the project of a radio frequency transmission system for the PA linearization, necessary for its operation into high input power levels, where signal distortions are not despicable. In this research, the applied model was the cascade between two Volterra series, which demand a parameter optimization step due to the nonlinear dependence between the block coefficients of the model. This paper investigates the influence of the damping choice for the Levenberg-Marquardt (LM) optimization method, into the relation between time elapsed and error returned by the optimization method. A LM derived method was developed looking for performance improvement. The new method achieved, for the analyzed set of values, results on average 0.45 dB better than the builtin MATLAB function *lsqnonlin*, with 38% of time reduction in the parameter identification step.**

***Index Terms*—Levenberg-Marquardt, power amplifier, behavioral modeling, Volterra series.**

I. Introduction

The transmission of data with fidelity and agility is the primary duty of a wireless communication system. In these systems, the power amplifier (PA) [1] is the principal device, responsible to increase the power of the signal to be transmitted. However a non-linearity is introduced by this device when it's performed into high power input levels due to the saturation of its internal transistors and the phase-shift caused by the input and output impedance matching networks. These non-linearities compromise its use, causing interference among neighbor channels and reducing its power efficiency. To fix this issue, usually the digital pre-distortion (DPD) [2] is employed.

Digital pre-distortion is a linearization method that consists into the previous application of a signal inverse to that one applied by the PA, resulting in a linear output response. For the project of a DPD, a behavioral modeling of the PA is requested, with high accuracy and low computational complexity. In this work the employed model was the cascade between two Volterra series [3].

The model is an improvement of the traditional Volterra series, a polynomial series with the capacity of reproducing non-linear responses and memory effects. Due to the concatenation of two Volterra series, the model becomes non-linear in its parameters, and then a method for their extraction and optimization is required. Heuristics methods should be avoided due to the accuracy trade off, which goes in opposition with our objectives. In [3], three methodologies for parameter extraction were explored: separable least-squares, iterative inverse problem solution and non-linear programming. In this work, the non-linear programming method will be applied. Among the available methods for non-linear parameter optimization there are two commonly employed: the Gauss-Newton and the Levenberg-Marquardt (LM) [4] methods.

This article contributes with the investigation of a non-linear parameter optimization method derived from LM, through the application of a damping switchable parameter, using finite-difference approximations of the Jacobian matrix for performance improvement [5].

II. The Levenberg-Marquardt method

The Levenberg-Marquardt method derives from the Gauss-Newton method, using linear approximations of the objective function for the problem optimization. The difference between the methods is in the introduction of the λ parameter, called Levenberg-Marquardt parameter or *damping* parameter. The method consists into the calculation of d_k, a gradient decrease direction in the function domain, through the relation:

$$(J^T(x_k)J(x_k) + \lambda_k I)d_k = -J^T(x_k)R(x_k) \tag{1}$$

where $R(x)$ is the objective function, $J(x)$ is the Jacobian matrix of R, $J^T(x)J(x)$ is an approach of the Hessian matrix of R and d is the gradient decrease direction. The damping parameter was included into the Gauss-Newton method to turn the Hessian approach always positive defined, and then reversible. The LM parameter is computed as:

$$\lambda_k = \frac{g_k^T v_k}{f(x_k)} \tag{2}$$

for an appropriate choice of v_k, where $g_k = J^T(x_k)R(x_k)$ and $f(x) = \frac{1}{2}\|R(x)\|^2$. Some choices for λ_k present in literature

are reported in Table I, while in Table II, a simplified algorithm of the generic LM method is presented.

TABLE I
DAMPING PARAMETER ACCORDING WITH THE v_k CHOICE

Method	λ_k	v_k
P1 [4]	$\frac{\|g_k\|^2}{f_k}$	g_k
P2 [4]	$\|g_k\|^2$	$f_k g_k$
P3 [4]	$\|g_k\|$	$\frac{f_k}{\|g_k\|} g_k$
P4 [6]	$\|R(x_k)\|^2 = 2f_k$	$\begin{cases} \frac{2f_k^2}{p(g_k)_i}, & \text{if } (g_k)_i \neq 0 \\ 0, & \text{if } (g_k)_i = 0 \end{cases}$
P5 [7]	$\|R(x_k)\| = 2\sqrt{f_k}$	$\begin{cases} \frac{2\sqrt{f_k^3}}{p(g_k)_i}, & \text{if } (g_k)_i \neq 0 \\ 0, & \text{if } (g_k)_i = 0 \end{cases}$
P6 [8]	$\lambda_k = \|g_k^T v_k\|$	$\max\left\{(v_{k-1})_i, \left\|\frac{\partial f(x_0)}{\partial x_i}\right\|\right\}$
P7 [9]	$\frac{2\|g_k\|}{3k}$	$\frac{2f_k}{3k\|g_k\|}, k > 0$

TABLE II
PSEUDO-CODE OF THE LEVENBERG-MARQUARDT METHOD

Algorithm: Levenberg-Marquardt
Given $x_0 \in \mathbb{C}^n$, define $k = 0$ While the stop conditions are not satisfied: Compute λ_k Get d_k, solution of $(J^T(x_k)J(x_k) + \lambda_k I)d_k = -J^T(x_k)R(x_k)$ $x_{k+1} = x_k + d_k$ $k = k + 1$ End

A comparative analysis was performed by [9], comparing the parameters of the methods P1-P7 through the application of a set of problems proposed by [10], which proved the best effectiveness and robustness of the P7 damping parameter in the time domain.

The software MATLAB has a builtin function, *lsqnonlin*, which implements the classic LM method with an adaptive approach for the damping parameter, proposed by [4]:

Given $x_0 \in \mathbb{C}^n$, $\lambda_0 > 0$, α and $\beta > 1$, compute d_k as in (1),

$$\textbf{if } R(x_k + d_k) < R(x_k) : x_{k+1} = x_k + d_k, \lambda_{k+1} = \frac{\lambda_k}{\alpha}$$

$$\textbf{else: } \lambda_k = \lambda_k \cdot \beta, \text{recompute } d_k$$

The values utilized by the lsqnonlin function are $\lambda_0 = 0.01$ and $\alpha = \beta = 10$.

III. IMPLEMENTED ALGORITHM

The algorithm was implemented with focus in computation time reduction and local convergence increasing. In this section will be presented the characteristics of the implemented code that contributes to achieve these goals and the pseudo-code of the final algorithm.

A. Algorithm characteristic 1: Derivative-free

Due to the size of the target problem (function domain dimension of the order of dozens or hundreds), the time for the Jacobian computation becomes a problem. To contour this, the approximation of the Jacobian matrix using finite-difference was chosen.

Be $\Delta R(x, h)$ a matrix where the element in position (i,j) is given by:

$$R_i(x + he_j) - R_i(x)$$

then, for h_k sufficient small, we can consider the approximation:

$$J(x_k) = \frac{1}{h_k}\Delta R(x_k, h_k)$$

and then (1) becomes equivalent to:

$$(\frac{1}{h_k^2}\Delta R(x_k, h_k)^T \Delta R(x_k, h_k) + \lambda_k I)d_k = -\frac{1}{h_k}\Delta R(x_k, h_k)R(x_k) \quad (3)$$

whose solution is the vector:

$$d_k = -h_k(\Delta R(x_k, h_k)^T \Delta R(x_k, h_k) + h_k^2 \lambda_k I)^{-1}\Delta R(x_k, h_k)R(x_k) \quad (4)$$

B. Algorithm characteristic 2: Damping Switchable

To increase the convergence of the method, the code was programmed to switch the damping parameter when a stop condition would be satisfied, searching for another decrease direction. The main parameter P7 was chosen as the main due to its better performance in time and robustness. The change into the damping parameter can occur up to five times according with the performance profile, in which:

$$(\lambda_k)_i = P_i \text{ , where } P = \{P7, P3, P4, P5, P6, P2\}$$

The damping parameter P1 was not implemented because, according with [9], it was not able to resolve any problem in a shorter time.

C. Algorithm characteristic 3: Stop conditions

The stop conditions of an optimization method are usually the number of iterations, $k > k_{max}$, and the gradient of the objective function, $g_k < g_{tol}$.

Commonly in the nearby of a local minima the algorithm is trapped on a direction that returns a very small decrease per iteration. To avoid this problem, it was established an absolute and relative error decrease stop condition:

$$\|R(x_k)\| - \|R(x_{k+1})\| < R_{tol} \text{ or } \frac{\|R(x_k)\| - \|R(x_{k+1})\|}{\|R(x_k)\|} < R_{tol} \quad (5)$$

These conditions cause the switch of the damping parameter, avoiding directions that do not contribute with a significant decrease and, at the end of the method, avoiding unnecessary iterations.

D. Pseudo-code of the final algorithm

The proposed algorithm explained in this section will be referenced from now on as P8. Table III presents a pseudo-code of the P8 algorithm, which is the sum of contributions of the three characteristics explained in the previous subsections.

TABLE III
PSEUDO-CODE OF THE P8 ALGORITHM

Algorithm: P8 Levenberg-Marquardt

Given $x_0 \in \mathbb{C}^n$, h_k, g_{tol}, $R_{tol} \in \mathbb{R}^+$, define $k = 0$, $i = 0$
While Eq. 5 is satisfied, $g_k < g_{tol}$, $k < k_{max}$, $i < 6$:
  $i = 0$
  DO:
    $P = P_i$
    Compute λ_k
    Get d_k, solution of Eq. 3
    $x_{k+1} = x_k + d_k$
    IF Eq. 5 is not satisfied and $i < 6$:
      $x_{k+1} = x_{k+1} - d_k$
    $i = i + 1$
  WHILE Eq. 5 is not satisfied and $i < 6$
  $k = k + 1$
End

IV. CASE STUDY: PARAMETER IDENTIFICATION OF A POWER AMPLIFIER BEHAVIORAL MODEL

In this section the cascade model developed in [3] is presented and numerical experiments are performed in order to compare the method P8 with the P7 method and the classic LM method (with adaptive damping parameter) inbuilt in the *lsqnonlin* MATLAB function.

A. The Volterra series cascade model for the PA behavioral modeling problem

A Volterra series is a polynomial series with the ability of reproducing memory effects. For the radio frequency power amplifier, the low-pass equivalent Volterra series is a convenient adoption, having the form [11]:

$$\hat{y}(n) = \sum_{p=1}^{P} \sum_{q_1=0}^{M} \sum_{q_2=q_1}^{M} \cdots \sum_{q_p=q_{p-1}}^{M} \sum_{q_{p+1}=0}^{M} \sum_{q_{p+2}=q_{p+1}}^{M} \cdots \sum_{q_{2p-1}=q_{2p-2}}^{M} \times \hat{h}_{2p-1}(q_1, \cdots, q_{2p-1}) \prod_{j_1=1}^{p} \hat{x}(n-q_{j_1}) \prod_{j_2=p+1}^{2p-1} \hat{x}^*(n-q_{j_2}) \quad (6)$$

where $\hat{x}(n)$ and $\hat{y}(n)$ are the complex-valued envelopes, respectively, at the input and output of the PA, $(^*)$ denotes the complex conjugate operator, M is the memory length, $P_0 = 2P - 1$ is the polynomial order truncation and $\hat{h}_{2p-1}(q_1, \cdots, q_{2p-1})$ are the low-pass equivalent Volterra kernels.

As the number of parameter in (6) increases quickly with M and P, a cascade model was proposed in [3], returning results that prove the best approximation and the complexity reduction of the cascade model. Figure 1 illustrates the cascade model, where $X1$ and $X2$ are matrices composed by the product presented in (6) for all index combinations, $H1$ and $H2$ are the coefficient vectors of each series and $I(n)$ and $y_{est}(n)$ are the output respectively of the first and second blocks.

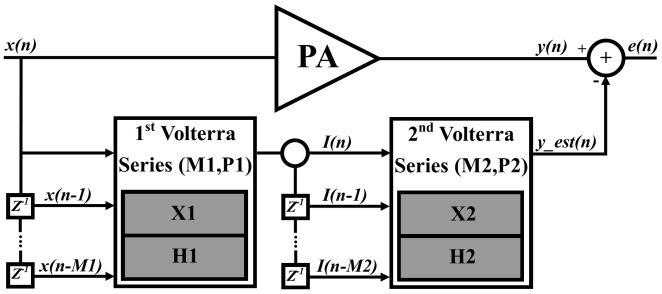

Fig. 1. Block diagram of two Volterra series in cascade [3]

B. Numerical Experiments

For the nonlinear parameter extraction method, an initial point is required for the method. In [3], the initial point was taken by the application of the separable least-squares method (SLS). In this work, ten random points were generated and applied into the three optimization functions. The time elapsed by each optimization was stored and the normalized mean-square error (NMSE) was computed for all cases, with relation to a validation dataset, as:

$$NMSE = 10log \frac{\sum_{n=1}^{N} (y\_est(n) - y(M1 + M2 + n))^2}{\sum_{n=1}^{N} y(M1 + M2 + n)^2} [\text{dB}] \quad (7)$$

The training and validation data were previously collected by [12], with a class AB PA with GaN based technology, excited by a 900 MHz carrier and stimulated by a WCDMA signal with 3.84 MHz of bandwidth. The choice of the cascade parameters was done based in the results obtained in [3]. The selected values were $[M1, P1, M2, P2] = [2, 3, 2, 1] \equiv 84$ coefficients, which is the first set of parameters that returned a better result than the best traditional model, with 244 coefficients.

The method P7 was implemented as in [9], with stop conditions $k_{max} = 150$ and $g_{tol} = 1e-4$. The method P8 was limited by the same stop conditions, in addition to the stop condition explained in Subsection III-C, with $R_{tol} = 1e-3$. The *lsqnonlin* MATLAB built-in function was also limited by $k_{max} = 150$ and $g_{tol} = 1e-4$.

Figure 2 presents the results obtained from this experiment. Both methods, P7 and *lsqnonlin*, reached the maximum of 150 iterations. In the first case, error increase can be observed in the middle of the method, however, if a stop condition was implemented, then the final error would be significantly higher, as presented in Table IV.

The difference between the methods in Table IV is exactly the contribution of the "damping switchable" implementation. The results were obtained from the application of the Volterra cascade model, with parameters $[M1, P1, M2, P2] = [1, 4, 1, 4] \equiv 80$ coefficients, into 80 random initial points. Table IV presents the five best results of the method P8 and the five worst results of the method P7 with insufficient decreasing stop condition. For both methods, $k_{max} = 150$, $g_{tol} = 1e-4$ and $R_{tol} = 1e-4$.

TABLE IV
EXPLICIT CONTRIBUTION OF THE 'DAMPING SWITCHABLE' IMPLEMENTATION

P7 + insuficcient decreasing				P8			
$\|R(x_k)\|$	$\|\nabla R(x_k)\|$	Iterations	NMSE [dB]	$\|R(x_k)\|$	$\|\nabla R(x_k)\|$	Iterations	NMSE [dB]
1.39E+00	1.67E+00	51	-29.77	1.28E+00	5.83E-01	97	-30.19
5.69E+00	2.31E+01	32	-17.70	1.30E+00	5.41E+00	87	-29.80
3.20E+00	9.01E+00	28	-22.85	1.30E+00	1.02E+00	110	-29.88
1.39E+00	7.19E+00	73	-29.19	1.38E+00	1.36E+00	77	-29.31
2.67E+00	1.99E+01	37	-24.08	1.39E+00	3.99E+00	74	-29.73
3.63E+04	7.99E+05	31	54.48	1.18E+01	8.17E+00	98	-11.09
1.01E+06	2.94E+06	13	74.40	8.80E+00	5.23E+00	51	-13.28
5.80E+06	1.31E+07	7	98.05	2.38E+00	1.29E+01	79	-25.50
1.08E+07	2.81E+07	15	108.47	1.05E+01	3.67E+00	68	-9.10
6.85E+08	9.47E+08	8	131.05	1.70E+00	8.08E-01	142	-27.93

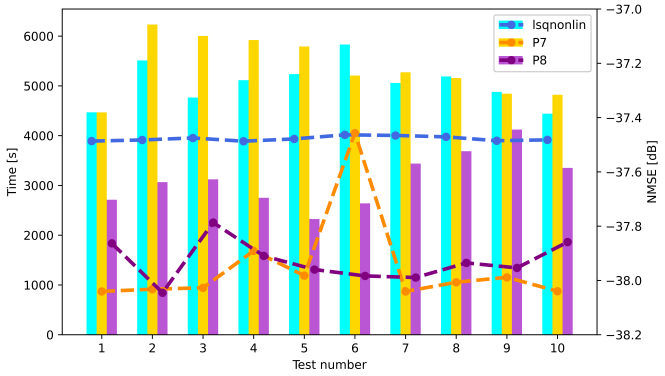

Fig. 2. Comparison of the requested optimization time and NMSE between three LM optimization methods for ten samples

V. CONCLUSIONS

The analysis of the results evidences that the proposed optimization method is faster in relation to the *lsqnonlin* MATLAB function and the simple LM method with damping parameter P7. The time reduction was, on average, about 38% in relation to the *lsqnonlin* function and 32% for the P7 method. This improvement was a consequence of the absolute and relative error decrease stop condition adoption, which avoided iterations that do not contribute significantly. However, this simple improvement does not contribute for the algorithm convergence as seen in Table IV, which highlights the need for a "damping switcher" that will avoid the premature collapsing of the optimization method and improve its local convergence. With the "damping switchable" implementation, the algorithm reached results, on average, of -0.45 dB in relation to the *lsqnonlin* function and $+0.02$ dB for the P7 method. The improvement returned by the P8 method turned it better for problems such the creation of NMSE curves, which demand for the training of thousands (or millions) individual model instances, where the total optimization time was a big problem.

ACKNOWLEDGMENT

The authors would like to acknowledge the financial support provided by National Council for Scientific and Technological Development (CNPq).

REFERENCES

[1] H. Wang, K. Sengupta, *"RF and Mm-Wave Power Generation in Silicon"*. Elsevier Science, 2015.
[2] P. B. Kenington, *"High Linearity RF Amplifer Design,"* Norwood. MA: Artech House, 2000.
[3] F. P. Ribeiro, E. G. Lima, *"Behavioral modeling of radio frequency power amplifiers using cascades of two Volterra series"*, 37th South Brazil Microelectronics Symposium, SIM2022, 2022.
[4] K. Levenberg, *"A method for the solution of certain problems in least squares"*, Quart. Ap. Math. 2, 1944, pp. 164–168.
[5] K. M. Brown e J. E. Dennis Jr. *"Derivative free analogues of the Levenberg-Marquardt and Gauss algorithms for nonlinear least squares approximation"*. Numerische Mathematik 18.4 (1971): 289-297.
[6] J. Fan e Y. Yuan, *"On the convergence of the a new Levenberg-Marquardt method"*. Technical Report, AMSS, Chinese Academy of Sciences, Beijing, China, 2001.
[7] N. Yamashita e M. Fukushima, *"On the rate of convergence of the Levenberg-Marquardt method"*. Technical Report 2000-008, Kyoto University, Kyoto, Japão, 2000.
[8] J. J. Moré, *"The Levenberg-Marquardt algorithm: implementation and theory. Em Proceedings of the 1977 Dundee conference on numerical analysis,"* Lecture Notes in Mathematics 630. G. A. Watson. (ed.) Berlin, Springer, 1978, pp. 105–116.
[9] K.A. Benatti, *"O Método de Levenberg-Marquardt para o Problema de Quadrados Mínimos não Linear."* Dissertação de mestrado, UFPR, 2017.
[10] J. J. Moré, B. S. Garbow e K. E. Hillstrom, *"Testing unconstrained optimization software"*. ACM Trans. Math. Software 7, 1981, pp. 17–41.
[11] S. Benedetto, E. Biglieri, and R. Daffara, *"Modeling and performance evaluation of nonlinear satellite links – a Volterra series approach,"* IEEE Trans. Aerosp. Electron. Syst., vol. 15, no. 4, pp. 494–507, Jul. 1979.
[12] A. T. Fukamati, E. G. Lima, *"Modelagem comportamental de amplificadores de potência utilizando cascata de dois polinômios com memória."* In: 2020 Iberchip Workshop, 2020, San Jose, Costa Rica. 2020 Iberchip Workshop, 2020. p. 1-4.