Selective Algorithm for Group Method of Data Handling Applied to Power Amplifier Modeling

ANA PAULA PRINCIVAL MACHADO
dept. Electrical Engineering (UFPR)
Universidade Federal do Paraná
Curitiba, Brazil
anaprincival@ufpr.br

EDUARDO GONÇALVES DE LIMA
dept. Electrical Engineering (UFPR)
Universidade Federal do Paraná
Curitiba, Brazil
elima@eletrica.ufpr.br

***Abstract*—Power amplifiers (PAs) are electronic devices commonly used in telecommunications that need to transmit information with high energetic efficiency. For this, it is necessary to use data manipulation methods that assist in the linearization of the output signal. This work presents two codes constructed based on the Group Method of Data Handling and which differ in their way of selecting the best coefficients to be used in the calculations of the neural network. The first method, called Embracing, assumes greater availability of data, while the second, called Selective, selects information from the beginning of the code. The comparison between the two logics shows that the most selective and computationally more complex structure, when searching for linearization since from the first layers, expresses minor errors and the best results in the output, becoming a reasoned option for use in PAs.**

Keywords—Power amplifiers, data manipulation methods, Group Method of Data Handling.

I. INTRODUCTION

A communication system consists basically of three elements: transmitter, medium and receiver [1]. In an amplifier, the transmitter converts the information into a form suitable for sending, delivering signal at a minimum energy level.

Power amplifiers (PAs) are devices that basically have the function of amplifying the capacity of the signal applied to their input. They are divided into classes, indicating how much the output signal varies within an operating cycle for a complete input cycle [2].

To keep linearity between the output signal and input signal, radio frequency PAs need to operate in regions with low input power. However, in order to obtain a system with good energetic efficiency, avoiding the exacerbated consumption of battery in isolated systems, operations in non-linear regions (in which greater power is required) are used, but, for this, it is necessary to appropriate methods that guarantee the linearization of the system [3].

There are different types of modeling for systems linearization, and among these neural networks stand out. The Feed Forward topology consists of a network in which data travel in only one direction, from the entrance to the last layer. Each layer is made up of neurons, which manipulate the data according to their programming. Its nomenclature is an analogy to biological neurons, which have potential for cell passage [3].

When it comes to data modeling that has many coefficients, some identification algorithms, such as method of Least Squares (LS), start to show inaccuracies in their results. In order to find an efficient method that produces the correct data collection, the work addresses Group Method of Data Handling (GMDH), which consists of algorithms capable of identifying, organizing and regressing systems with a large number of inputs.

II. GMDH

The GMDH model was initially introduced by Ivakhnenko in the late 1960s with the aim of studying non-linear relationships between input and output variables [4]. It has consolidated itself as a pioneering method in the matter of self-organization, acting with minimal human intervention and presenting accurate final results, even with a large number of inputs. It proved to be of great importance in the field of electronics and telecommunications by assisting in the linearization of transmission chains [3].

This modeling method is organized in layers, each with a certain number of neurons. Each neuron receives two inputs and results in only one output, which will be used as an input to a next layer using all possible combinations, as shown in Fig. 1.

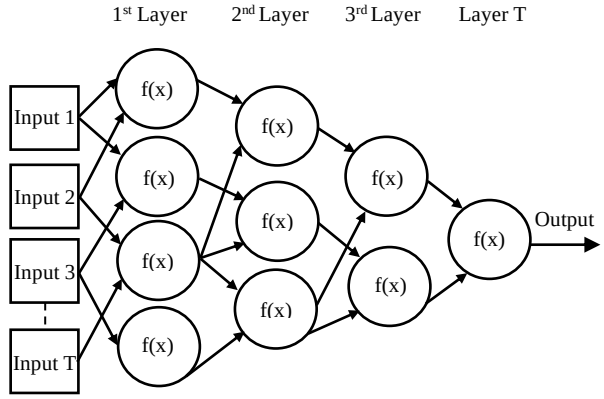

Fig. 1 – Block diagram of GMDH

The number of neurons in each layer is directly defined by the number of real inputs, which, starting from the second layer, is made up of the real outputs of the previous neurons [4]. Each of these neurons has the activation function explained by

$$f(x) = m + nx_i + ox_j + px_ix_j + qx_i^2 + rx_j^2, \quad (1)$$

where the index i subscribed refers to the first input of the neuron, and the index j refers to the second input. The coefficients m, n, o, p, q and r are adjustable and particular to each neuron, directly influencing the output of each neuron and, therefore, in the calculation of the next layer.

The main points of the method are to draw the topology of the network: number of inputs, number of layers and the best neurons of each one, also analyzing the most advantageous combinations for the accuracy of the final result. The identification and manipulation of the adjustable coefficients of each activation function are also essential for the greater mastery of modeling with GMDH.

III. GMDH MODEL ADAPTED FOR BEHAVIORAL MODELING OF PAS

When using the GMDH model for the behavioral modeling of PAs it is necessary to adapt the activation function contained in each neuron, since the involved signals are of complex values (real and imaginary parts representing the envelope signal), used for greater accuracy in modeling and reduced complexity, especially when used in telecommunications [3].

For the neuron to process inputs from the complex domain, its activation function must be complex, respecting the condition of being contained in a limited and complex domain [5]. The activation function previously presented is modified to

$$f(x) = mx_i + nx_j + ox_i|x_i| + px_j|x_j| + qx_i|x_j| + rx_j|x_i|, \quad (2)$$

where m, n, o, p, q and r are adjustable coefficients, i and j remain references to the positions of the inputs, this time complex. From the third term onwards, the complex value module is represented by $|x|$, being one of the factors in the multiplication of variables.

When GMDH is used for PA modeling there is only one output at the end of the neural network, which depends on the input applied in the current and past moments. The amount of past instants used is directly influenced by the initial number of inputs (E), ranging from 0 to (E – 1).

In order to always obtain one output, the last layer of the network must converge to just one neuron. In the present work, three layers are used, chosen for simplicity of presentation and clarity of behavior, but the model can be adapted for a much higher number of layers.

The number of neurons in the first layer depends directly on the initial entry of inputs, given by $Cr_1 = a!/[b!(a - b)!]$, where Cr_1 is the resulting number of neurons, a is the total number of inputs to be applied and b the number of inputs for each neuron (commonly 2, according to the basic model of Ivakhnenko). As each neuron results in only one output, these become the inputs for the second layer. So, for the second layer the amount of neurons is $Cr_2 = Cr_1!/[b!(Cr_1 – b)!]$, and for the third layer is $Cr_3 = Cr_2!/[b!(Cr_2 - b)!]$, following the logic of using the outputs of layer 2 as inputs for layer 3. This mathematical rule follows for the rest of the neural network.

It is important to realize that for layer 1 there are two possible scenarios. The first, exemplified in Fig. 2, admits that only three neurons are used in calculating the network, since one is repeated in the direction to the second layer. The Fig. 3 shows the second scenario, which admits the use of four different neurons. Therefore, the code works with these two propositions, analyzing the order and combination of all possibilities by repeating the use of one neuron, and then analyzes the order and combination admitting the use of all four neurons. After all the comparisons made respecting this criterion, the smallest error is obtained with the identification of the best neurons used in calculation.

The modeling consists of choosing the best neurons to be kept in each layer: which is the only neuron to be kept in third layer, which two neurons to be used in the second layer and which is the most strategic selection in the first layer.

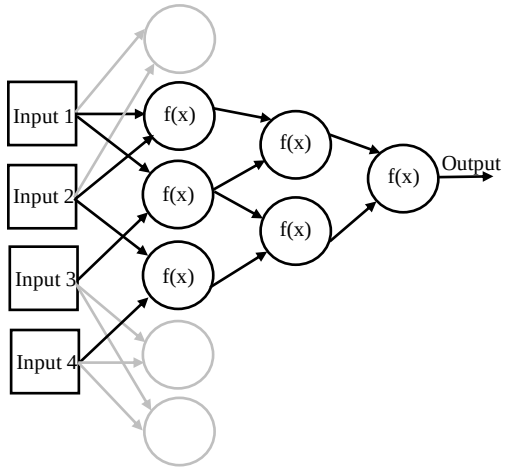

Fig. 2 – First scenario using three neurons in first layer

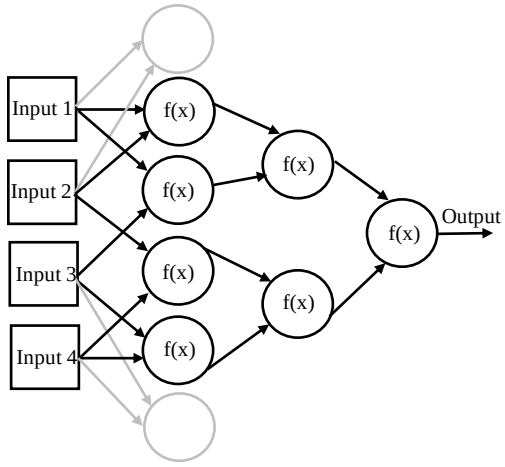

Fig. 3 – Second scenario using four neurons in first layer

IV. DESIGN OF THE BEHAVIORAL MODEL OF PA BASED ON THE GMDH

The GMDH model is considered non-linear in its parameters, since its adjustable coefficients are multiplied by

one another, reaching powers greater than 1. However, a linear in its parameter model can be obtained if the coefficients of each layer are calculated independently. The work consists of obtaining the coefficients of the first layer and, from these, extracting the coefficients of the second layer and, finally, calculating the coefficients of the last layer, analyzing each result in isolation to estimate the next outputs.

The coefficients of each layer are extracted using LS, a method used for linear regression that consists of minimizing the sum of squares of errors [6]. To define the most strategic neurons (those with the best coefficients), all possible combinations in the neural network are tested. At the end of each layer, the Normalized Mean Square Error (NMSE) is calculated, used to access the accuracy of the results, defined by

$$NMSE = 10\log \frac{\sum_{n=1}^{N} |e(n)|^2}{\sum_{n=1}^{N} |yref(n)|^2}, \quad (3)$$

where N is the total number of samples and n is the referenced instant. The term $e(n)$ represents the difference between the desired output (indicated by $yref(n)$) and the estimated output. The best selection of the neural network is the one with the lowest NMSE, in all the methods presented here.

The project works with two modeling methods that use Mean Square Error (MSE) on each layer for the extraction of the coefficients, differing in the scope of the selection of neurons.

The first method, here called Embracing, assumes the availability of all neurons in the first layer, with only one extraction of coefficients at that time. In the same way, it is admitted that all neurons of the second layer are used, and this way the coefficients are also extracted by only a calculation of MSE. As for the third layer, it is assumed that there is only one neuron, and the calculation of the MSE is performed according to the number of possibilities of combinations, given by the same value of Cr_2, that is the number of neurons in the second layer. This happens because two inputs are strictly required in the single neuron of the last layer, so all combinations of 2 to 2 of the available neurons in the second layer are tested, directed to each routine by code construction modeling.

The second method, called Selective, begins to select neurons from the beginning of the code. It is assumed that in the first layer only the neurons that will actually be used for the second layer are present, which may be in the amount of 3 or 4. When it is admitted that there is a neuron repetition, totaling 3 neurons in the first layer, there are tested n_3 possibilities, where n_3 is given by triple the number of combinations from 3 to 3 of the total neurons of the first layer, Cr_1. In this calculation, the number of neurons in the first layer is chosen as the domain of the combinations because it is the one that will leave the other selections for the rest of the neural network; the set of 3 to 3 is selected by the direct number of neurons to be used – in this case, one is repeated; and the final multiplication by 3 is performed in order to analyze all possible positions assumed by each neuron, as shown in Fig. 4. When four different neurons are used, the number of tested possibilities is n_4, where n_4 is given by triple the number of combinations from 4 to 4 of the Cr_1 value, a calculation whose variables are justified in the same way as for when there is repetition, as shown in Fig. 5. Therefore, the total number of extractions is n_{total}, the result of the sum of possibilities n_3 and n_4.

Still on the Selective method, the second layer has only two fixed neurons, each receiving two inputs and resulting in only one output. Then, the coefficients are obtained from each neuron only once, totaling two extractions. Finally, in the third layer there is only one neuron, which receives two inputs and results in an output, with only an extraction of coefficients, which are the most significant of the neural network.

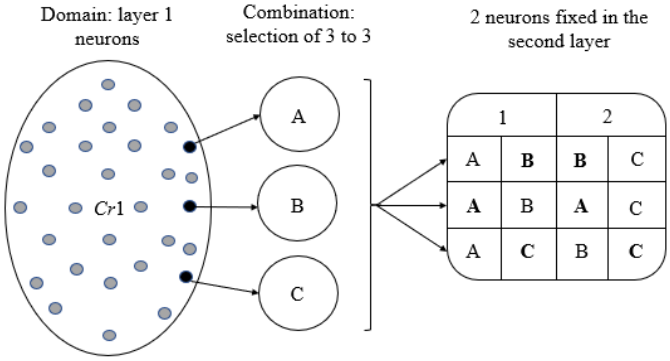

Fig. 4 – Selective method with neuron repetition

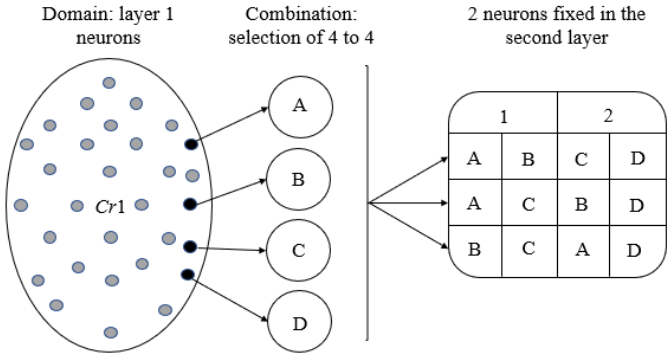

Fig. 5 – Selective method without neuron repetition

V. Simulation Results

At this point, the Embracing and Selective methods described in Section IV will be compared.

The number of inputs for the neural network varied from 3 to 6, and Table 1 shows the extraction and validation NMSE side by side within each method.

Table I – Comparison between methods

	EMBRACING		SELECTIVE	
Inputs	**NMSE Ext.**	**NMSE Val.**	**NMSE Ext.**	**NMSE Val.**
3	-31.37	-31.27	-31.37	-31.27
4	-31.08	-31.68	-34.89	-35.39
5	-30.55	-30.84	-36.12	-36.66
6	-27.66	-27.74	-36.53	-37.17

The proximity between the results of extraction and

validation proves that there are no problems of noise modeling or ill-conditioning (poor conditioning of the regression matrix), resulting in good results for the codes. However, it is noticeable that the Selective method is outstanding, since it is computationally more complex than the Embracing method. When making MSE calculation a greater number of times, it results in more accurate results, since it presents the smallest NMSE.

Still looking at Table I, the Selective method is considered the best in terms of cost-benefit, based on its most assertive form of selection and the presentation of the smallest errors. The NMSE, measured on the decibel scale, doubles the error order every 3 units of difference, which reinforces the relevance of this method.

By varying the *E* to higher values, the complexity of the calculations and accuracy of results increase, so for a greater number of inputs the number of extractions can vary almost exponentially. The conclusion of the work demands an empirical analysis, considering the advantages of carrying out an additional large number of tests for an improvement of NMSE not so significant. Therefore, considering the performance of the tests and the comparative analysis of results, the best routine performed is the one for 5 inputs applied to the Selective code, as it presents one of the smallest NMSE – with very close extraction and validation – and does not require such complexity to estimate the best coefficients and neurons to be used.

Fig. 6 shows a comparison between the estimated and desired output amplitudes using the Selective method with the application of 5 inputs as the initial network number. About forty samples were selected to facilitate the visualization of the accuracy of the method. No clear differences between measured and estimated amplitudes are visible, confirming the high accuracy of the designed GMDH.

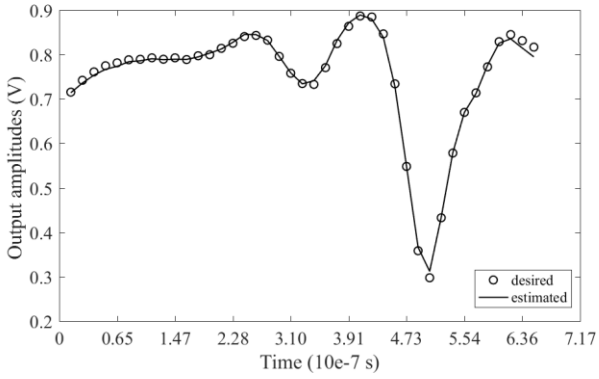

Fig. 6 – Comparison between output amplitude waveforms

For these tests, the MATLAB software was used, in which both codes were built, chosen for its objectivity in terms of modeling. Several commands offered by the program were used, among them the "\" to perform the LS. The samples used are classified as *floating-point double precision* and were taken ready at the beginning of the work – the main purpose was the data modeling, and it was not worked with their extraction –, obtained by a class AB PA that employs a high electron mobility semiconductor manufactured in GaN technology. A 900 MHz carrier frequency was used, modulated by a WCDMA 3GPP envelope signal with a bandwidth of approximately 3.84 MHz. For the measurement of input and output information, a vector signal analyzer of the Rohde & Schwarz FSQ type was used, with 61.44 MHz as the sampling frequency [7]. The extraction samples are 3,221 in size, while the validation samples are in the order of 2,001.

VI. Conclusions

When realizing the need for linearization of PA output signals, the study of data manipulation methods is also turned to its application in the field of telecommunications. The work presented the results obtained from the construction of two programming codes based on the GMDH model that search to transfer signals through low energy levels. These differ in the way they select the best coefficients to be maintained in the neural network: one is more embracing than the other.

After carrying out several tests, the most selective method stands out, which makes a more careful analysis about the neurons and outputs to be maintained at each layer and, therefore, presents better final results and minor errors. Its logic is applicable to use in PAs, since the solutions resulting from the applied inputs are all concrete.

However, the code uses a large number of combinations to perform all calculations. In order to reduce the complexity, it is suggested that future works focus on a method that reduces the number of calculation possibilities by defining a smaller amount of strategic neurons for each layer.

ACKNOWLEDGMENT

The authors would like to acknowledgment the financial support provided by National Council for Scientific and Technological Development (CNPq) and Fundação Araucária de Apoio ao Desenvolvimento Técnico Científico e Tecnológico do Estado do Paraná under the Program PIBITI UFPR 2020.

References

[1] HAYKIN, S. An introduction to Analog and Digital Communications. [S.I.]: Wiley, 2006.

[2] CRIPPS, S. RF power Amplifiers for Wireless Communications, 2nd edition. Norwood, MA: Artech House, 2006.

[3] ZANELLA, A. F.; LIMA, E. G. A GMDH based approach for the Behavioral Modeling of Radiofrequency Power Amplifiers. In: 19o Simpósio Brasileiro de Micro-ondas e Optoeletrônica e 14o Congresso Brasileiro de Eletromagnetismo, 2020. 19o Simpósio Brasileiro de Micro-ondas e Optoeletrônica e 14o Congresso Brasileiro de Eletromagnetismo. p. 266-270.

[4] IVAKHNENKO, A. G. The group method of data handling in long-range forecasting. Technological Forecasting and Social Change,v. 12, 2-3 1978.

[5] KIM, T.; ADALI, T. Fully complex multi-layer perceptron network for nonlinear signal processing. Journal of VLSI Signal Processing Systems for Signal, Image, and Video Techology, v. 32, 1-2 2002. ISSN 13875485. DOI: 10.1023/A : 1016353216361.

[6] MATHEWS, J.; SICURANZA, G. 2000. Polynomial signal processing. John Wiley & Sons, Inc.

[7] BONFIM, E.; LIMA, E. G. A modified two dimensional Volterra-based series for the low-pass equivalent behavioral modeling of RF power amplifiers. Progress In Electromagnetics Research M, Vol. 47, 27-35, 2016.