Real-valued Neural Network Based in 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 devices that amplify the signal applied to their input, and it is necessary that they do so with low energy use. In order to guarantee this, the signal manipulation must occur in a linear way, and situations involving higher orders need the application of linearization methods. Among these methods, neural networks can be mentioned, where the Group Method of Data Handling (GMDH) model stands out. Based on it, two programming logics were built that differ in their criteria for selecting neurons to be kept in the layers of the network. Then these logics were adapted to work with real-valued data, thus becoming real neural network structures. At the end of the processing of each sample, the Normalized Mean Square Error (NMSE) is calculated, reaching better results of -35.27 dB in extraction and -35.98 dB in validation. The good performance of the built topologies makes them efficient options in data linearization.**

Keywords—Power amplifiers, real neural network, Group Method of Data Handling.

I. Introduction

Power amplifiers (PAs) are devices usually used in telecommunications that basically amplify the signal applied to their input. It is necessary that they demand low energy from the system in which they operate, and for that the PA signal must be processed in linear means [1].

However, in specific cases that require more power, the signal reaches higher orders, and thus linearization methods are used to maintain low energy consumption [2]. These methods can be approached in a logical-mathematical way, and some of them even involve the use of neural networks. Among the models of existing neural networks, those of the Feed Forward type stand out, which process data in a unidirectional way from input to output. In this topology, there is, for example, the Group Method of Data Handling (GMDH), which has a high capacity for organization and accuracy of results even with a large amount of information applied to the input [3].

The present work approaches the GMDH topology and uses it as a basis for the construction of two programming logics, which aim to linearize the signal coming from the PA. These structures differ in the way of neural organization, one being more rigorous than the other regarding the selection of data that will cross the layers of the network. Such structures called Embracing and Selective were discussed in [4] and their Expanded versions based on Laguerre basis functions were introduced in [5]. Considering that in [4] and [5] all codes were optimized to work with complex data, the contribution of this work is to adapt the original Embracing and Selective structures to work with real data. So, after the redefinition of input possibilities and manipulation of complex samples, organizational variations are tested according to the information entered in the input. At the end of each routine the Normalized Mean Square Error (NMSE) is calculated, which makes it possible to identify the best structure of the neural network, now real.

II. Modeling PAs behavior based on GMDH

The GMDH model was initially developed by the Ukrainian professor Ivakhnenko in the mid-1960s, while studying linearization of input and output data. The model stood out for being a self-organizing neural network, capable of producing accurate results even when processing a lot of information [3]. The GMDH network is organized into layers, each of which has a certain number of neurons. Once the efficient modeling of the GMDH is known, it is up to the designer to adapt it according to his objective.

In the present work, the GMDH network was adapted to work with data from PAs, studying its behavior when processed by neural networks with three layers. The topology was designed so that the second layer has two neurons, while the third layer has only one. The number of neurons in the first layer is given by

$$Cr_1 = \frac{E!}{2!(E-2)!}, \tag{1}$$

where Cr_1 is the number of neurons in layer one and E is the number of inputs inserted into the network. The number of outputs of the layer is the same number of neurons present in it. Furthermore, the activation function contained in each one is given by

$$f(x) = m + nx_1 + ox_2 + px_1x_2 + qx_1^2 + rx_2^2, \tag{2}$$

where the subscripted indices *1* and *2* refer to each neuron input. The coefficients m, n, o, p, q and r are extracted in each layer and become essential, since they are used to produce the inputs of the next layer and also to study the validation of the entire structure.

A. *Embracing and Selective codes*

In [4] two codes were presented based on GMDH and that differ in the way of selecting the inputs of the layers, one being more rigorous than the other. The stricter one is called Selective – selecting neurons from the first layer –, while the second is called Embracing and selects only from the second layer. Both were trained with data from a PA and presented good results, becoming concrete options for signal linearization [4], and in the present work the Embracing and Selective codes are used in a real neural network.

The Embracing code assumes the availability of all neurons in the first layer, extracting the coefficients through the Mean Square Error (MSE) only once. Then the network goes to a second layer with only two neurons, which select inputs from a set containing Cr_2 possibilities, calculated by

$$Cr_2 = \frac{Cr_1!}{2!(Cr_1 - 2)!}. \quad (3)$$

Thus, at each value of E, the Embracing trains a certain number of routines in order to find the best neurons that will compose the neural network. The number of routines is calculated by combining 2 to 2 (number of neurons in the second layer) and Cr_2 (number of available possibilities). Due to the adaptation to real data – which increases the complexity of the training – the possibilities of organization in relation to the original topology were reduced [4]. Fig. 1 shows how the Embracing method is organized.

On the other hand, the Selective method selects neurons from the first layer, without the intermediate set Cr_2. However, this method has two scenarios: repeating and non-repeating.

In the first scenario, it is assumed that 3 neurons are selected from the Cr_1 set, in which the output of one of them is repeated as an input in the second layer. So, the number of times the network is trained in the scenario with repetition is determined by the combination of 3 to 3 of Cr_1, illustrated by Fig. 2.

In the second scenario, it is assumed that 4 neurons are selected from the Cr_1 set, and thus there are 4 distinct inputs in the second layer. In this way, the number of times the network is trained in the no-repeat scenario is determined by the combination of 4 to 4 of Cr_1, as shown in Fig. 3. The total number of code routines for each value of E is given by the sum of trainings from both scenarios.

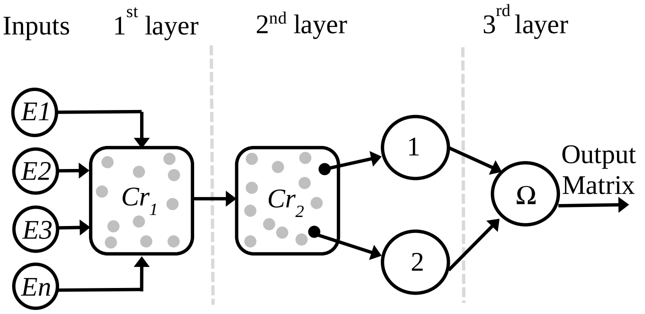

Fig. 1 – Embracing design

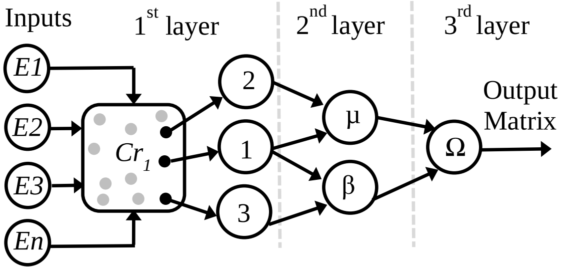

Fig. 2 – Scenario with repetition of the Selective

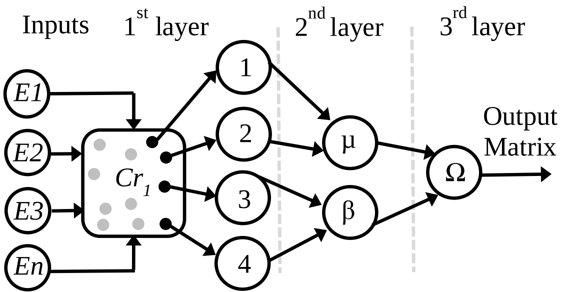

Fig. 3 – Scenario without repetition of the Selective

B. *Adaptation and training for a real network*

When working with real networks, it is necessary to adapt the source dataset – often complex – so that all samples assume the real form.

The modeling is done in baseband, so it always uses complex numbers. So, starting from a set of complex samples from a PA, its elements are called complex data. To obtain real data, this original complex set is manipulated and divided into its real and imaginary parts, as will be described in detail. The model that uses only real data is composed by two networks: the network that estimates the real part is called In-phase (I) Network, while the network that estimates the imaginary part is called Quadrature (Q) Network. At the end of all the processing, the analyzes are all done with complex numbers, and so the data that are as real components will be manipulated again to assume their complex form.

In details, the real topology presented in this work operates as follows: starting from a complex set, manipulations are made to form real matrices with all possible inputs of the GMDH. So, two independent GMDH networks were created: the In-phase (I) Network and the Quadrature (Q) Network [2], which are essentially the Embracing or Selective codes. The difference between them is the desired output, where in the In-phase Network it is defined by $b_n cos(\alpha_n - \theta_n)$, while in the Quadrature Network it is given by $b_n sin(\alpha_n - \theta_n)$, where b is the modulus of the output sample, α is the angle of the output sample, θ is the angle of the input sample and n is the referenced instant of each one of them.

These two networks aim to separately process the data set, whose input – the same for both – is already in real form. At the end of processing, the output of each network is manipulated to form a complex number again and, with it, extract the NMSE and thus verify the accuracy of the project. The Fig. 4 shows a diagram to illustrate the general functioning of the network.

To build the input matrices, the variable M is defined, which will adjust the number of columns in the matrix, where each column represents a possibility to be inserted in the GMDH. Given an original vector in complex form, increasing M will increase the number of possibilities when referencing the magnitude and angle of samples with increasingly past instants. For example, for $M = 1$ there are four inputs: a_n, a_{n-1}, $cos(\theta_n - \theta_{n-1})$ and $sin(\theta_n - \theta_{n-1})$, where a is the modulus of the input complex sample, θ is the angle of the input complex sample and n is the sample instant. For $M = 2$, there are seven inputs: a_n, a_{n-1}, a_{n-2}, $cos(\theta_n - \theta_{n-1})$, $sin(\theta_n - \theta_{n-1})$, $cos(\theta_{n-1} - \theta_{n-2})$ and $sin(\theta_{n-1} - \theta_{n-2})$. And so the organization rule holds for any value of M.

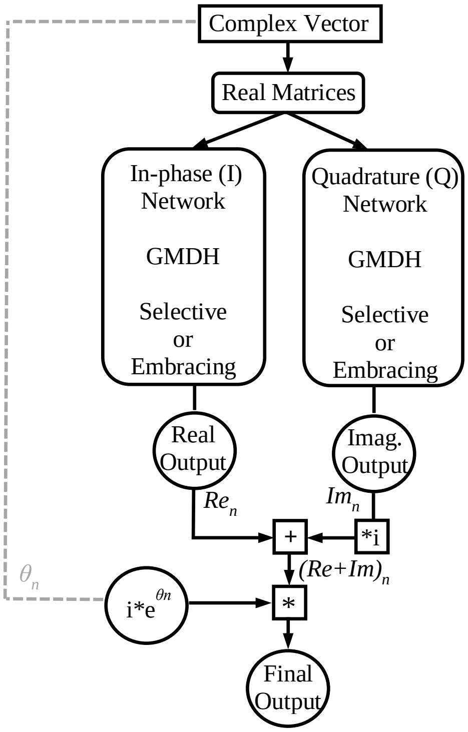

Fig. 4 – General network operation

Therefore, each value of E has the number of possibilities of inputs offered by each value of M. With this, the GMDH network is trained h times, where h is given by the number of combinations of E in E and the number of columns in the matrix of actual inputs.

The code is designed to store the output vector of each routine in an array. Thus, when all the routines for an M and an E previously chosen are completed, there are two final output matrices with h columns: one for the In-phase Network and another for the Quadrature Network.

Then a third code works to combine the real and imaginary columns, assembling a complex number and enabling the calculation of the NMSE, which represents the accuracy of the structure, so the smaller its value, the smaller the error and the more accurate the calculation. Thereby, by identifying the best extraction NMSE, it is also possible to identify which value of h and which combination of output matrices led to this result, and with these data the validation NMSE is calculated. It is defined by

$$NMSE = 10\log \frac{\sum_{n=1}^{N} |e(n)|^2}{\sum_{n=1}^{N} |yref(n)|^2}, \tag{4}$$

where $yref(n)$ is the desired output at each instant, $e(n)$ is the difference between the desired output and the estimated output at each instant, and N is the total number of samples.

III. Simulation Results

All the presented programming logics were built in the MATLAB software, using the mathematical tools offered by it [6]. The extraction of data from the PA had already been done previously, not being the main purpose of the present work [7]. These data are of floating-point double precision type, with 3,221 extraction samples and 2,001 validation samples, coming from a class AB PA manufactured in GaN technology, modulated by a WCDMA 3GPP envelope signal with 3.84 MHz of bandwidth and with 900 MHz carrier frequency. For the analysis of input and output information, a Rohde and Schwarz FSQ vector signal analyzer was used, which has a sampling frequency of 61.44 MHz [7].

In the simulations, the value of E was varied from 3 to 6, and the value of M was varied from 1 to 4. It is important to note that when $M = 1$ there are 4 possible input possibilities, and thus its use for $E = 5$ and $E = 6$ is not possible, since they demand more inputs than what is offered. Furthermore, the number of columns in the output matrix is different for the same M and the same E in the Embracing (only one scenario) and Selective (two scenarios) codes.

The results obtained with the Embracing and Selective codes when used as a real neural network are presented in Tables I to IV in which each one presents the data obtained with the variation of the value of M. From 1 to 4, each M offers four, seven, ten and thirteen possibilities of inputs, respectively. The tables show the variation of inputs, the number of trained routines and the NMSE (in decibels) of extraction and validation for each method.

The entire project was performed on a computer with an 11th generation Intel Core i5 processor, which has 6 cores and a base frequency of 2.4 GHz. It also has 8 GB RAM memory, SSD and Intel Iris X integrated video card. In it, for the extraction of coefficients with 3 inputs and $M = 1$, for example, the *tic toc* command of MATLAB indicates an execution time of 4.40 ms to perform MSE in the I network, and 3.75 ms to do the same in the Q one.

Table I - Results for $M = 1$

	EMBRACING			**SELECTIVE**		
Inputs	**Routines**	**NMSE Ext. (dB)**	**NMSE Val. (dB)**	**Routines**	**NMSE Ext. (dB)**	**NMSE Val. (dB)**
3	4	-26.00	-26.10	4	-26.00	-26.10
4	1	-25.98	-26.13	2	-26.95	-27.03

Table II – Results for *M* = 2

	EMBRACING			SELECTIVE		
Inputs	**Routines**	**NMSE Ext. (dB)**	**NMSE Val. (dB)**	**Routines**	**NMSE Ext. (dB)**	**NMSE Val. (dB)**
3	35	-28.22	-28.86	35	-28.22	-29.01
4	35	-27.54	-27.55	70	-30.38	-30.64
5	21	-27.50	-27.47	42	-32.17	-32.13
6	7	-27.70	-27.82	14	-32.17	-32.13

Table III – Results for *M* = 3

	EMBRACING			SELECTIVE		
Inputs	**Routines**	**NMSE Ext. (dB)**	**NMSE Val. (dB)**	**Routines**	**NMSE Ext. (dB)**	**NMSE Val. (dB)**
3	120	-29.50	-30.18	120	-29.50	-30.18
4	210	-31.38	-31.40	420	-31.68	-32.26
5	252	-30.71	-30.74	504	-34.74	-35.11
6	210	-30.72	-30.72	420	-34.74	-35.11

Table IV – Results for *M* = 4

	EMBRACING			SELECTIVE		
Inputs	**Routines**	**NMSE Ext. (dB)**	**NMSE Val. (dB)**	**Routines**	**NMSE Ext. (dB)**	**NMSE Val. (dB)**
3	286	-30.34	-31.02	286	-30.34	-31.02
4	715	-33.37	-33.77	1430	-34.38	-34.82
5	1287	-34.15	-34.66	2574	-35.27	-35.98
6	1716	-33.70	-34.38	3432	-35.27	-35.98

The presented data show that the results improve as the value of *M* increases. This behavior is observed because there are more input possibilities available, which allows a more efficient organization of the neural network. It is also noted that the variation of *E* for the same *M* causes a change in the values of NMSE, and the increase in the number of inputs does not necessarily imply a decrease in the error. Thereby, being an empirical analysis, the same input can be used in more than one neuron in the first layer, and therefore does not guarantee improvements.

It is important to point out that the best performances are concentrated in the highest values of *M*, but it is also when the codes require a much larger number of simulations to find the best organization. Thus, it is necessary to analyze the cost-benefit by the designer, aiming for the most accurate results through a more complex simulation. In this specific case study, the best trade-off is provided by applying 6 inputs in Selective method when *M* = 3. This case results in -34.74 dB as a NMSE of extraction, which is 0.53 dB highest than the best case (provided by applying 6 inputs to the Selective method when *M* = 4), and needs to run about 88% fewer routines than it.

In general, both structures present good performance and a similar behavior regarding the variation of *E* and *M*. In addition, it can be seen that the Selective code always presents a performance equal or superior to the Embracing, since it was developed with the objective of to be more accurate in your choice of neurons.

IV. Conclusions

Given the importance of maintaining linearity in the processing of the PAs signal, it is necessary to study data manipulation methods to guarantee this. Among these methods, it can be mentioned the neural networks, explored in this work.

Initially, two structures based on the GMDH model were built using programming logic, with the objective of manipulating the data from the PA and maintaining its linearization. These structures differ in the way they are selected for neurons, where one is more rigorous than the other, but both perform well.

In the work presented, these codes were adapted to operate with real data, being processed by two real neural networks that differ in the construction of the desired output. The targets of each network use the module and phase of each complex sample and, after training each routine, the accuracy of the project can be verified through the calculation of the NMSE. When achieving small errors in a wide range of organization routines, these codes are considered to reach the objective for which they were developed, becoming concrete options for data linearization.

Acknowledgment

The authors would like to acknowledge the financial support provided by National Council for Scientific and Technological Development (CNPq) under the Program PIBIC UFPR 2021.

References

[1] S. Cripps, RF power Amplifiers for Wireless Communications, 2^{nd} edition. Norwood, MA: Artech House, 2006.

[2] 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.

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

[4] MACHADO, A. P. P.; LIMA, E. G. . Selective Algorithm for Group Method of Data Handling Applied to Power Amplifier Modeling. In: XXI Microelectronics Students Forum, 2021, Campinas. Proceedings of the XXI Microelectronics Students Forum, 2021. p. 1-4

[5] MACHADO, A. P. P.; NYPWIPWY, V. B.; FRANÇA, C.; LIMA, E. G. Selective Algorithm for Expanded Group Method of Data Handling Applied to Power Amplifier Modeling. Accepted for publication in Journal of Integrated Circuits and Systems.

[6] C. Moler, Numerical Computing with MATLAB. Philadelphia: SIAM, 2004.

[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.