Approximation of Inverse Functions With and Without One-to-one Mapping

João Pedro Hinning, Eduardo Gonçalves de Lima
UFPR, Curitiba, Brasil
jphinning@gmail.com, eduardo.goncalves.lima@gmail.com

***Abstract*—This paper is concerned with digital predistortion, aiming to linearize Power Amplifiers (PA) with high accuracy and low computational work. First, we demonstrate how PAs can be modeled with a Volterra-based series and then how the PA response inverse function (f^{-1}) can help with linearization. After that, we establish and test our f^{-1} with memory and memoryless polynomials. The experimental results demonstrate that we can predict inverse functions accurately with both proposed approaches if we maintain the condition that the nonlinearities of the function are invertible.**

Index terms- Digital predistortion, linearization, power amplifier, memory polynomial.

I. INTRODUCTION

With the evolution of mobile communication, improving energy efficiency in the field is becoming increasingly important [1]. Wireless transmission requires wide bandwidth over a wide signal dynamic range and, in order to achieve signal transmission and high efficiency, RF power amplifiers (PAs) operate in their nonlinear region, causing amplitude and phase distortions [2].

A commonly used approach to deal with this problem and correct errors caused by it is to use a digital pre-distorter (DPD) that can compensate the net effect of the nonlinear behavior by obtaining the PA response curve inverse function (f^{-1}). Here we present a mathematical model that can approximate f^{-1} through a one-to-one mapping method, which means that the nonlinearities of the function are invertible and where the output is an explicit function of the input in the same instant [3]. Since the model used in this article is a polynomial function, our objectives are to investigate the necessary conditions for guaranteeing one-to-one mapping in polynomial inverse functions, approximating f^{-1}s with mathematical models, and applying these learned concepts in (DPD).

The paper is structured as follows: in Section II we review our mathematical model using a memory polynomial, then in Section III we analyze the concept of DPD. In Section IV, we explain the necessary conditions for obtaining f^{-1} in a polynomial without memory followed by Section V where we do the same with a memory-polynomial. Then, we report our simulation results in Section VI and finally present our conclusions in Section VII.

II. MEMORY POLYNOMIAL MODEL FOR PAS

In order to mathematically model our PA, we will be using the black box method [4], where we assess a system solely from the outside and are concerned only by its input/output behavior.

This approach is generally used when we have little to no knowledge of the PA internal circuit and, even though its accuracy is sensible to the model structure, it presents less computational complexity which is key for successful DPD. Among the options to model nonlinear PAs, the Volterra series stands out as a lightweight algorithm. It can be described as a linear function of its parameters, which means we can use simple techniques such as multiple regression to extract our output.

In a discrete Volterra series [5], to estimate the dynamic consequences of our response curve, the output at instant $\tilde{y}(n)$ depends on input $\tilde{x}(\mathrm{n})$ and past inputs $\tilde{x}(\mathrm{n}-\mathrm{m})$ where $m \geq 0$. This means that each input at any instant affects future outputs, but the influence it exercises on the output diminishes over time. Because of this, we generally limit the number of past input samples with an arbitrary variable called M. Moreover, to estimate the nonlinear outcome, the Volterra series output is described as a polynomial function of its inputs where the order of the function can be limited at P. An alternative to simplify the Volterra series is to use its particular instance called memory polynomial (MP) [6], which only keeps the terms involving inputs taken at a single sample. The MP is then described by the following equation:

$$\tilde{y}(n) = \sum_{p=1}^{P} \sum_{m=0}^{M} b_{2p-1,m} \, |\tilde{x}(n-m)|^{2p-2} \, \tilde{x}(n-m) \quad (1)$$

where $b_{2p-1,m}$ are complex coefficients and each one represents a different value. Observe that the accuracy of a memory polynomial is strongly tied with M and P values. The higher the values of these parameters, the greater the accuracy of the results. However, we need to be aware that the number of coefficients grows exponentially, which can result in ill-conditioning of the regression matrix, making our estimate results unreliable.

III. PREDISTORTION ANALYSIS

In Section II, we've established a model for a PA that depends on a previously collected dataset to predict new output. Furthermore, in order to tweak our predicted results, all we need to do is empirically test P and M parameters and find which value can have the best results with the lowest computational work. In this section, we discuss how we can use the inverse function of the PA response curve to correct nonlinear behavior as a method of DPD.

Two major blocks are present in a DPD system as shown in Figure 2. The DPD block applies the inverse function to our electric input $u(n)$. The PA block computes the result $x(n)$ of the previous operation, resulting in $y(n)$. With this system, we can have an idea of how to arrange the DPD, and PA equipment in the physical layer of a project.

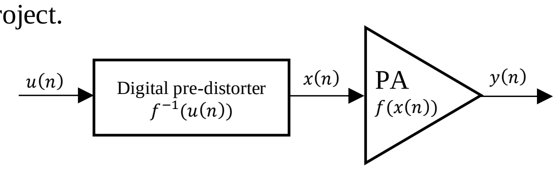

Fig. 1. *DPD block diagram*

After modeling the amplifier response curve and getting its inverse function, we can linearize the PA output by modifying the input of our system with f^{-1}. The result of this operation happens to shape our original $u(n)$ in a contrary direction, compensating for the PAs distortion. The more accurate our model is, the better $y(n)$ will be able to maintain its original information. Figure 3 illustrates the necessity of the inverse function in our system.

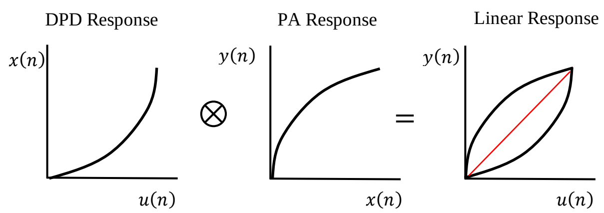

Fig. 2. *DPD, PA, and linear response characteristic function*

The inverse function present in the pre-distorter, in essence, consists of exchanging input and output roles in our model. This means that we can parametrize a response curve utilizing a memory polynomial, linear regression, and, after that, exchange its x and y, which will have us ending up with f^{-1}.

IV. MEMORYLESS AND NONLINEAR RESPONSE INVERSION THEORY

In this section, we analyze how to guarantee a one-to-one mapping of an inverse function for our PA model.

Firstly, we must have in mind that the necessary conditions for an f^{-1} to exist is that, in the range of interest, the original f has to be injective, meaning that one element of its input must correspond with a unique element of its output. In our case, since f will be represented by a polynomial of complex coefficients, the objective is not to find a perfect inverse function, but to try approximating optimal coefficients for any value of P, the parameter that dictates the order of our equation.

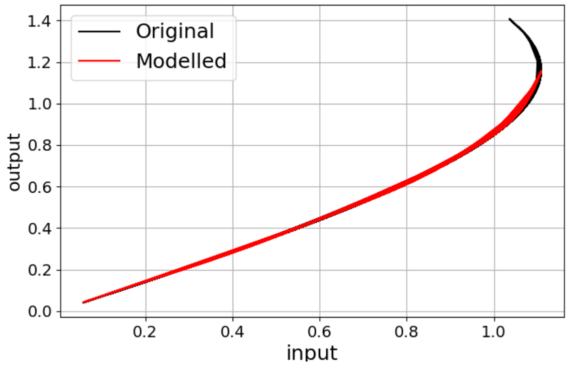

Fig. 3. *Generic comparison between desired inverse function and modeled response.*

In Figure 3, we can see that the better our curve fits the desired inverse function, the closer we get to a one-to-one mapping. However, the higher the value of our input, the more unwanted nonlinear behavior we get, which degrades de approximation of f^{-1}. Therefore, it should be noted that we only have a valid inverse function as long as the rate of change of our direct response is greater than zero $(\frac{dy}{dx} > 0)$. Furthermore, since we're dealing with a memoryless system, we need to emphasize that the above condition is valid for any instant, in other words, our input is only allowed to go up to a certain limit, or else it won't correspond to reality.

V. MEMORY POLYNOMIAL INVERSION THEORY

In Section IV, we've established that an inverse function without memory can be modeled until we reach a unique threshold in our input. In a memory polynomial, however, we need to treat our values beforehand to make sure they correspond with the original function.

A method for confirming the existence of a one-to-one mapping in our predicted inverse function consists in replacing past values with known ones. In an MP, this means substituting our previous samples at n-m (m = 1,2, ..., M) with a previously collected dataset. The following equation summarizes this change for the case M=1:

$$\widetilde{out}(i) = \sum_{p=1}^{P} b_{p,0}|in(i)|^{2p-2}\, in(i) + b_{p,1}|\widetilde{ext}(n-1)|^{2p-2}\, \widetilde{ext}(n-1) \quad (2)$$

In the coefficients $b_{p,n}$, each one represents a different value. They solely depend on the order of the function and n, which is the instant we're validating. $\widetilde{ext}(n)$ represents our extracted inputs from real PA measures and, therefore, they're known values. Our input $in(i)$ is a list of a hundred evenly spaced inputs starting at 0 and going until we reach $|\widetilde{ext}(n)|$. This equation will result in a different memoryless polynomial response for each instant *n* because the value of $\widetilde{ext}(n)$ changes through each iteration. As a consequence, we must test every single instant in our extracted dataset in order to evaluate a one-to-one mapping to our inverse function.

By modeling the equation, we can now assess if a specific instant is valid in our inverse function when we compare an $\widetilde{out}(i)$ that has a rate of change of zero or less ($\frac{dy}{dx} < 0$) with $|\widetilde{ext}(n)|$. If the module of our extraction is greater than our output value, then the instant won't be compatible with our inverse function. See how $|\widetilde{ext}(n)|$ is now our threshold, and its value changes depending on the instant we are testing. As a consequence, each instant should be assessed individually, and, if one sample fails to be satisfied, we can already claim that our function does not respect one-to-one mapping.

VI. Simulation Results

In this section, we will overview how we were able to simulate DPD in both memoryless and memory instances. For that, we will first describe our tools and detail our experimental data and then apply the principles explained in Sections IV and V, validating our methods with two datasets: one that has a one-to-one mapping and another that doesn't.

The proposed PA model has been implemented in Python with the libraries NumPy, for floating point double precision in matrix calculation and to apply the least squares method; Matplotlib for chart creation and visualization and SciPy to load a previously measured PA dataset. To calculate the response rate of change for both methods described in Sections IV and V, we simply compared the current output with the previous one. The derivative then would be equal to or less than zero as increasing input reduced output. We used a NumPy method to create a vector with a hundred evenly spaced values as described in Section V. We used the method's standard precision, float64, to assess our results.

To obtain the outcome presented in our simulations, it was utilized a PA manufactured with gallium nitride (GaN) technology, operating in class AB and excited by a carrier at a frequency of 900 MHz, modulated by a 3GPP WCDMA signal with 3.84 MHz of bandwidth [7]. The input and output data collected were measured using a Rohde & Schwarz FSQ VSA vector signal analyzer, with a 61.44 MHz sampling frequency. The data was divided into an extraction set and a validation set. The extraction one has 3,221 samples and the validation set has 2,001 samples.

A. Case study analysis without memory

For the validation of the memoryless instance, firstly it was created a polynomial using an MP with parameters P=2 and M=0 by applying the equation in our PA measured data. It generated the following coefficients: $[b_{1,0} = 0.960 - 0.016j; b_{3,0} = -0.108 + 0.027j]$

After that, in order to generate two samples for our experiment, we applied the same input extracted previously from a real PA with different gains in the regression matrix and got two different response curves. One with gain=1 and the other with gain=1.5. This part must be done so we can see one of the charts with perfect one-to-one mapping and the other failing to model the PA inverse. The method utilized here to validate one-to-one mapping is the same as described in Section IV. Consequently, by analyzing both response charts and their derivatives, we can conclude that the input limit established by the rate of change of the graph with gain=1 is its own maximum value of $1.45\,V$, and gain=1.5 is, approximately, $1.15\,V$.

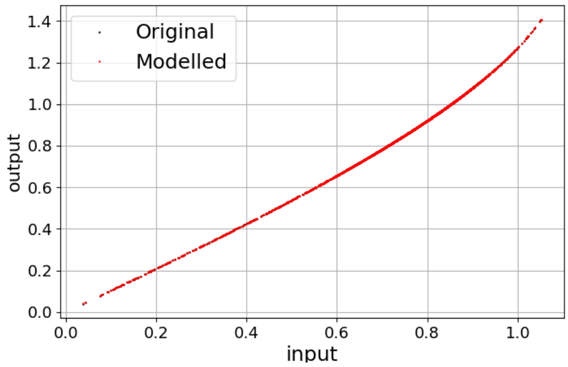

Fig. 4. *Inverse response with a one-to-one mapping*

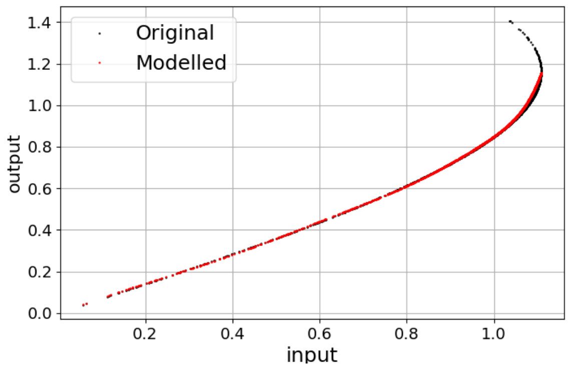

Fig. 5. *Memoryless inverse response without one-to-one mapping.*

After setting up our polynomials, we can now exchange their output and input values and create our ideal inverse response curves. With that done, we have now a new set of inputs and outputs that can be modelled by using equation (2) in Section IV. In our experiment, by testing the parameter values of P=2, 4, 7, and 10, the one that had the best accuracy/computational work ratio was P=7. Both inverse response charts can be seen in Figures 4 and 5:

In both charts, the modeled response is represented by red dots and the original inverse function by black dots. In Figure 4, where we had one-to-one mapping, we can clearly see that our model was accurate in predicting output values since we can barely see any black dots on the chart. On the other hand, in Figure 5, with no one-to-one mapping, the black region was poorly modeled and we can't predict anything above our limit input of 1.15 V.

B. *Case study analysis with memory*

In a similar way to Subsection VI.A, we first create a polynomial with our previously measured data with the parameters P=2 and M=1. It generates the following coefficients:

$$[\ b_{1,0} = 1.099 + 0.103j\ \ b_{1,1} = -0.166 - 0.128j$$

$$b_{3,0} = -0.317 - 0.019j;\ b_{3,1} = 0.240 + 0.054j\ ]$$

In order to validate what was discussed in Section V, we apply in our polynomial the same input, but with gain=0.65, chosen because it has a one-to-one mapping, and gain=1.5, without mapping. We verify each instant of both response curves by creating an array of a hundred values going from 0 to $|\widetilde{ext}(n)|$ as mentioned previously. By doing that, we can verify how many samples do not meet the established criteria and determine whether they are accurate to the inverse function or not. The first response curve with a 0.65 gain provided the following outcome in Figure 6.

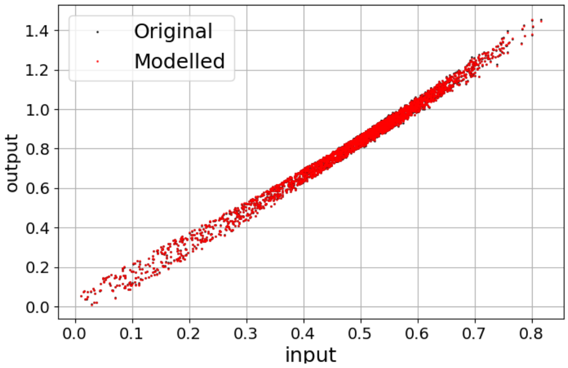

Fig. 6.*Memory polynomial response with gain = 0.65*

All tested instants were able to match the criteria. The second curve with a 1.5 gain, unlike the first, gave a much more inaccurate end result in Figure 7.

When modeling the inverse function for both gains, we also experimented with P and M values in order to determine the best accuracy/computational work ratio. In this case, P=5 and M=5 were the best combinations of values, as shown in Figures 6 and 7. In Figure 6 we can see that we can predict the output function accurately with the red dots, unlike in Figure 7, where all the black dots correspond to the number of instants that do not meet the criteria for one-to-one mapping. With that, it's clear that without a previous established one-to-one mapping our prediction methods become less accurate, even with high computational work.

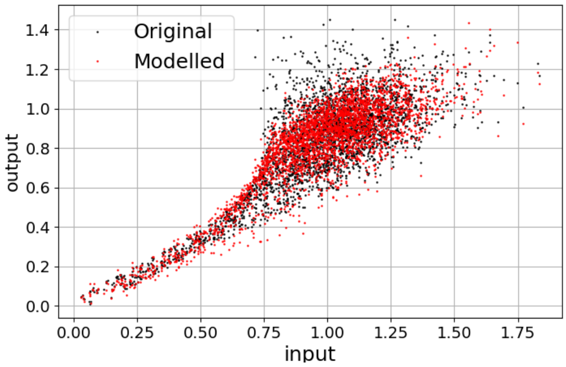

Fig. 7.*Memory polynomial response with gain = 1.5*

In this instance, 1,130 of 3,221 instants failed the criteria, meaning that 35% of our output is inaccurate.

VII. Conclusion

In this paper, it was established and tested four ways of handling a MP inverse function, by assessing memory and one-to-one mapping. We've concluded that the $f^{-1}s$ could be accurately predicted when we had a previous established one-to-one mapping with both memory and memoryless polynomials. On the other hand, when we detach the invertibility of our $f^{-1}s$ by changing the gain of the datasets, our results become a lot more inaccurate.

Acknowledgments

The authors would like to acknowledge 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 PIBIS UFPR 2021.

References

[1] T. Huang, W. Yang, J. Wu, J. Ma, X. Zhang, D. Zhang. A Survey on Green 6G Network: Architecture and Technologies. IEEE Access – Multidisciplinary open access journal, 7(4), 175758-175768 (2019).

[2] S. C. Cripps, "RF Power Amplifiers for Wireless Communications", ArtechHouse, 2006.

[3] P. B. Kenington, High Linearity RF Amplifier Design. Norwood, MA: Artech House, 2000.

[4] L. Ljung, System Identification: Theory for the User. Englewood Cliffs, NJ: Prentice Hall, 1999.

[5] M. Schetzen, "Nonlinear system modeling based on the Wiener theory," Proc. IEEE, vol. 69, no. 12, pp. 1557–1573, Dec. 1981.

[6] J. Kim and K. Konstantinou, "Digital predistortion of wideband signals based on power amplifier model with memory," Electron. Lett., vol. 37, no. 23, pp. 1417–1418, Nov. 2001.

[7] E. J. Bonfim and E. G. Lima. "A modified two dimensional volterra-based series for the low-pass equivalent behavioral modeling of RF power amplifiers." Progress In Electromagnetics Research M, v. 47, p. 27-35, 201