Coefficient estimation of nonlinear high power amplifiers using LMS/RLS adaptive algorithms

Mridula Malhotra[1*], Amandeep Singh Sappal[2]

[1]Research Scholar , ECE Department, Punjabi University, Patiala (Pb)- India

[2]Assistant Professor, ECE Department, Punjabi University, Patiala (Pb)- India

Abstract

The linearization problem of high power amplifier (PA) used in modern wireless communication system is a tedious task. As the nonlinearity of power amplifier is best removed using digital predistorter (DPD), the nonlinear coefficients of PA are the pre-requisite. In this paper, the adaptive algorithm (LMS / RLS) for Volterra kernels estimation have been applied for the nonlinear coefficient identification. A third order nonlinear system has been analyzed mathematically to implement these adaptive algorithms.

***Keywords**: digital predistorter (DPD); least mean square (LMS) algorithm; power amplifier (PA); recursive least square (RLS) algorithm*

1. Introduction

In the modern communication system such as OFDM and QAM, it is easy to get an efficient and less complex method than the traditional ones to compromise the multipath fading; however, they have a highly variable envelope and require more energy consumption. To reduce this energy consumption of communication systems, several high-efficiency PA have been proposed [1]. However, PAs are highly nonlinear device and have memory effects, when employed with amplitude modulated wideband signals [2]. In order to reduce the distortions, the digital predistortion is the most preferable and effective solution due to its high flexibility and simple working. DPD also has nonlinear characteristics, but has exactly reciprocal amplitude in a way that when it is cascaded with PA, it shows linear characteristics [3]. The nonlinearity of a system can be defined by various models, e.g. Volterra model and its reduced forms like memory polynomial (MP) model and its variants. Various models like Hammerstein, wiener, memory polynomial etc. and their hybrid are applied on power amplifier and digital predistorter by researchers to mathematically model the nonlinearity of the system [4-9]. The nonlinear coefficients of power amplifier are firstly identified by coefficient estimation method and are used to generate a predistorter having exactly reciprocal amplitude characteristics such that the cascade of PA and DPD generate linear characteristics at the output.

The LMS (least mean square) and RLS (recursive least square) adaptive algorithms are widely used to identify the nonlinear coefficients of power amplifiers. In LMS adaptive algorithms output is proportional to the product of input to the predistorter and output error. LMS algorithms execute quickly but converge slowly. This drawback is eliminated in RLS algorithm.

2. Coefficient Estimation using Least Square Methods

A third order nonlinear system with memory is identified using the adaptive algorithm (LMS / RLS) for Volterra kernels estimation. The implementation of the adaptive Volterra filter is based on the extended input vector and on the extended filter coefficients vector. Due to the linearity of the input-output relation of the Volterra model with respect to filter coefficients, the implementation of the adaptive algorithm was realized as an extension of the algorithm for linear filters.

The first order input vector, corresponding to a filter length $M = 3$, is defined as follows:

$$X^{(1)T} = \left[x(n)\, x(n-1)\, x(n-2) \right] \tag{1}$$

If we consider equal memories for different orders filters, “the second order input vector” can be expressed by:

$$X^{(2)} = X^{(1)} * X^{(1)T} \tag{2}$$

For symmetric kernels only the elements $x_{i,j}$, having $i \geq j$, of $X^{(2)}$, are selected in the input-output relation of the Volterra filter. Hence “the second order input vector”, written in vector form is:

$$X^{(2)T} = \begin{bmatrix} x^2(n)\, x(n) * x(n-1)\, x(n) * x(n-2)\, x^2(n-1) \\ x(n-1) * x(n-2)\, x^2(n-2) \end{bmatrix} \tag{3}$$

and has the dimension (1×6).

For "the third order input vector" we propose to express the multiple input delayed signal products by matrices elements. These matrices can be generated by multiplying “the second order input vector" defined according to Eq. (2) by the elements of the first order input vector. If we consider equal filters, $M=3$, and symmetric kernels it follows:

$$X^{(2)} * x(n) = \begin{bmatrix} x^3(n) & x^2(n) * x(n-1) & x^2(n) * (n-2) \\ \cdots & x(n) * x^2(n-1) & x(n) * x(n-1) * x(n-2) \\ \cdots & \cdots & x(n) * x^2(n-2) \end{bmatrix} \tag{4}$$

$$X^{(2)} * x(n-1) = \begin{bmatrix} \cdots & \cdots & \cdots \\ \cdots & x^3(n-1) & x^2(n-1) * x(n-2) \\ \cdots & \cdots & x(n-1) * x^2(n-2) \end{bmatrix} \tag{5}$$

$$X^{(2)} * x(n-1) = \begin{bmatrix} \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & x^3(n-2) \end{bmatrix} \quad (6)$$

Hence, "the third order input vector" consists in fact, in that case, of 3 matrices as indicated in Equations and (4) - (6)corresponds to a symmetric third order Volterra kernel. We can write "the third order input vector" in vector form as follows:

$$X^{(3)T} = \begin{bmatrix} x^3(n)x^2(n)*x(n-1)\ldots x(n)*x^2(n-2)x^3(n-1) \\ \ldots x(n-1)*x^2(n-2)x^3(n-2) \end{bmatrix} \quad (7)$$

Its dimension is (1×10).

The defined input vectors will be used to implement the LMS and RLS Volterra filter in a typical nonlinear system identification application.

3. Volterra Kernels Estimation by the LMS Adaptive Algorithm

A typical adaptive technique employing LMS algorithm used for Volterra kernels identification is shown in Figure 1. The Volterra filter of fixed order and fixed memory adapts to the unknown nonlinear system using one of the various adaptive algorithms. A simple and commonly used algorithm uses an LMS adaptation criterion [10]. The aim of this section is to discuss the simplest of the algorithms, the LMS algorithm. Although the LMS algorithm has its weaknesses, such as its dependence on signal statistics, which can lead to low speed or large residual errors, it is very simple to implement and well behaved compared to the faster recursive algorithms.

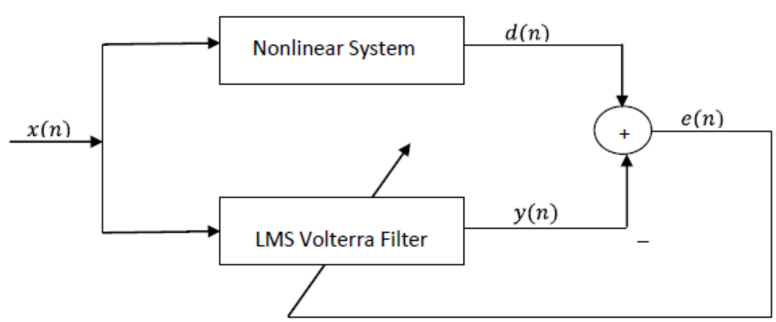

Figure 1. Volterra kernel identification using adaptive method

The main topic of this section is to discuss the extension of the algorithm to the nonlinear case using the previously defined input vectors. The discrete time impulse response of a first order (linear) system with memory span *M*, is written in vector form as in Eq. (8) and the input vector as in Eq.(9).

$$H_k^{(1)T} = \left[h_k^{(1)}(0)\,h_k^{(1)}(1)\ldots h_k^{(1)}(M-1)\right] \quad (8)$$

$$X_k^{(1)T} = \left[x_k \quad x_{k-1} \quad \cdots \quad x_{k-M+1} \right] \tag{9}$$

In Eq.(9) the filter order is written as superscript. This notation will be kept consistent for the rest of the section. Then, the output of a linear system is written as:

$$y_k = H_k^{(1)T} * X_k^{(1)} \tag{10}$$

At sample k, the desired output is d_k and the linear adaptive filter output is y_k . For the LMS algorithm, we minimize the Eq.(11).

$$E\left[e_k^2 \right] = E\left[d_k - H_k^{(1)T} * X_k^{(1)} \right] \tag{11}$$

The vector H^* that minimizes the Eq. (11) is given by :

$$H^* = R_{xx}^{-1} * G$$

Where: $R_{xx} = E\left[X_k^{(1)} * X_k^{(1)T} \right]$ is the input correlation matrix and $G = E\left[X_k^{(1)} * y_k \right]$. The well known LMS update equation for a first order filter is:

$$H_{k+1}^{(1)} = H_k^{(1)} + \mu * e_k * X_k^{(1)} \tag{12}$$

Where μ is a small positive constant (referred to as the step size) that determines the speed of convergence and also affects the final error of the filter output.

The extension of the LMS algorithm to higher order (nonlinear) Volterra filters involves a few simple changes. Firstly the vector of the impulse response coefficients becomes the vector of Volterra kernels coefficients. Also the input vector, which for the linear case contained only a linear combination, for nonlinear Volterra filters, complicates.

Consider the Volterra representation with symmetric kernels. There are two parts of this representation: (1) the Volterra kernel estimates, and (2) the products of the delayed input signal. If we express the Volterra kernels and the input signal products in vector form, then we can write the adaptive Volterra filter output using the vector notation. Each Volterra kernel (estimate at sample *k*) can be written in vector form.

For simplicity we have constructed the nonlinear adaptive filter considering only first order and 3rd order Volterra kernels.

The Eq.(13) gives "the input matrix" at sample *k*, containing the first, second and the third order input vectors defined previously.

$$X_k = \begin{bmatrix} X_k^{(1)T} \\ X_k^{(2)T} \\ X_k^{(3)T} \end{bmatrix} \tag{13}$$

The size of the input matrix is determined by the size of the third order input vector $X_k^{(3)}$. “The filter coefficients matrix” at sample k is given by:

$$H_k = \begin{bmatrix} H_k^{(1)T} \\ H_k^{(2)T} \\ H_k^{(3)T} \end{bmatrix}$$

Where $H_k^{(1)T}$ is given by the Eq. (12), $H_k^{(2)T}$ and $H_k^{(3)T}$ are the second and third order kernel expressed in vector form as indicated in Eq.(14) and (15) respectively.

$$H_k^{(2)T} = \left[h_k^{(2)}(0,0) \quad h_k^{(2)}(0,1) \quad \dots \quad h_k^{(2)}(M-1, M-1) \right] \tag{14}$$

$$H_k^{(3)T} = \left[h_k^{(3)}(0,0,0) \quad h_k^{(3)}(0,0,1) \quad \dots \quad h_k^{(3)}(M-1, M-1, M-1) \right] \tag{15}$$

The update equation for the LMS Volterra filter can be written also in matrix form:

$$\begin{bmatrix} H_{k+1}^{(1)T} \\ H_{k+1}^{(2)T} \\ H_{k+1}^{(3)T} \end{bmatrix} = \begin{bmatrix} H_k^{(1)T} \\ H_k^{(2)T} \\ H_k^{(3)T} \end{bmatrix} + e_k * \begin{bmatrix} \mu_1 & 0 & 0 \\ 0 & \mu_2 & 0 \\ 0 & 0 & \mu_3 \end{bmatrix} \begin{bmatrix} H_k^{(1)T} \\ H_k^{(2)T} \\ H_k^{(3)T} \end{bmatrix} \tag{16}$$

In the nonlinear case it is possible to set different step sizes for different order kernels. Consequently we have introduced the step size matrix M, defined by

$$\mu = 1 \begin{bmatrix} \mu_1 & 0 & 0 \\ 0 & \mu_2 & 0 \\ 0 & 0 & \mu_3 \end{bmatrix} \tag{17}$$

4. Volterra Kernels Estimation by the RLS Adaptive Algorithm

The Volterra filter of a fixed order and a fixed memory adapts to the unknown nonlinear system using one of the various adaptive algorithms. The use of adaptive techniques for Volterra kernel estimation has been well studied. A simple and commonly used algorithm is based on the LMS adaptation criterion. Adaptive Volterra filters based on the LMS adaptation algorithm are computational simple but suffer from slow and input signal dependant convergence behavior and hence are not useful in many applications. The aim of this section is to discuss the efficient implementation of the RLS adaptive algorithm on a third order Volterra filter. Due to the linearity of the input-output relation of the Volterra model with respect to filter coefficients, the implementation of the RLS algorithm can be realized as an extension of the RLS algorithm for linear filters. Hence we define the extended input vector, for a third order Volterra filter, as:

$$X = \begin{bmatrix} x(n)\dots x(n-M+1)\,x^2(n)\,x(n)*x(n-1)\dots \\ x^2(n-M+1)\,x^3(n)\,x^2(n)*x(n-1)\dots x^3(n-M+1) \end{bmatrix} \tag{18}$$

and the extended filter coefficients vector as:

$$H = \begin{bmatrix} h(0)\dots h(M-1)\,h(0,0)\,h(0,1)\dots h(M-1,M-1) \\ h(0,0,0)\,h(0,0,1)\dots h(M-1,M-1,M-1) \end{bmatrix} \tag{19}$$

The elements of the extended input vector can be easily actualized based on the first order, second order and third order input vectors using the proposed relations (4) - (6). As in the linear case the adaptive nonlinear system minimizes the following cost function at each time:

$$J(n) = \sum_{k=0}^{n} \lambda^{n-k} * \left(d(k) - H(n) * X'(k)\right)^2 \tag{20}$$

Where $H(n)$ and $X(n)$ are the coefficients and the input signal vectors, respectively, as defined in (18) and (19), λ is a factor that controls the memory span of the adaptive filter and $d(k)$ represents the desired output. The solution of equation (20) can be obtained recursively using the RLS algorithm.

The RLS algorithm updates the filter coefficients according to the following steps:
I Initialization: *Define the filter memory length for* $H(n)$ *and* $X(n)$

$H(0) = [0 \quad 0 \quad \dots \quad 0]$;

Where $C_{XX}(0) = \delta * I$ is a small positive constant.

II. Operations: for an iteration (n)

1. Create the input vector: $X(n)\,2$

2. Compute the error:

$$e(n/n-1) = d(n) - H(n-1) * X'(n)$$

3. Compute the scalar:

$$\mu(n)\,X(n) * C_{XX}(n-1) * X'(n)$$

4. Compute the matrix:

$$G(n) = \left(C_{XX}(n-1) * H(n-1)\right) / (\lambda + \mu)$$

5. Updates the filter vector:

$$H(n) = H(n-1) + e(n/n-1) * G'(n)$$

6. Updates the matrix

$$C_{XX} = \lambda^{-1} * \left(C_{XX}(n-1) - G(n) * X(n) * C_{XX}(n-1)\right)$$

In the relations above C_{XX} denotes the inverse autocorrelation matrix of the extended input signal. Inversion was done according to the matrix inversion lemma.

5. Conclusion

The PA performance is affected by the nonlinearity and memory effects. So, PA output depends on the input characteristics like frequency, signal bandwidth, peak-to-average power ratio, type of load, and type of modulation. The volterra series is widely applied to capture these effects; however, it requires complex identification procedures. Thus, transparent white box fitting procedure like memory polynomial model and its variants are mostly used to capture PA behavior and can be easily employed with larger system simulation. In this paper, the LMS/RLS optimization algorithm has been used to improve the model performance by finding the nonlinear coefficient of the system.

References

[1] Cripps, S.C., 2006. RF Power Amplifiers for Wireless Communications, Artech House Microwave Library (Hardcover), Norwood, MA, USA, Artech House, Inc.

[2] Ghannouchi, F.M., Hammi, O. and Helaoui, M., 2015. Behavioral modeling and predistortion of wideband wireless transmitters, John Wiley & Sons.

[3] Gharaibeh, K.M., 2011. Nonlinear distortion in wireless systems: Modeling and simulation with MATLAB, John Wiley & Sons.

[4] Kim, J. and Konstantinou, K., 2001. Digital predistortion of wideband signals based on power amplifier model with memory. Electronics Letters, 37(23), 1417-1418.

[5] Manjula, S. and Selvathi, D., 2015. Optimal design of low power CMOS power amplifier using particle swarm optimization technique. Wireless Personal Communications, 82(4), 2275-2289.

[6] Bipin, P.R., Rao, P.V. and Aruna, S., 2018. A new pre-distorter for linearizing power amplifiers using adaptive genetic algorithm. In International Proceedings on Advances in Soft Computing, Intelligent Systems and Applications. Springer, Singapore, pp. 403-413.

[7] Safari, N., Roste, T., Fedorenko, P. and Kenney, J.S., 2008. An approximation of Volterra series using delay envelopes, applied to digital predistortion of RF power amplifiers with memory effects. IEEE Microwave and Wireless Components Letters. 18(2), 115-117.

[8] Morgan, D.R., Ma, Z., Kim, J., Zierdt, M.G. and Pastalan, J., 2006. A generalized memory polynomial model for digital predistortion of RF power amplifiers. IEEE Transactions on signal processing. 54(10), 3852-3860.

[9] Sappal, A.S., Patterh, M.S. and Sharma, S., 2011.Fast complex memory polynomial-based adaptive digital predistorter. International Journal of Electronics. 98(7), pp. 923-931.

[10] Haykin, S. and Widrow, B. eds., 2003. Least-mean-square adaptive filters (Vol. 31). John Wiley & Sons.