A RECURSIVE LEAST SQUARES ALGORITHM WITH REDUCED COMPLEXITY FOR DIGITAL PREDISTORTION LINEARIZATION

Saijie Yao *Hua Qian* *Kai Kang* *Manyuan Shen*

ABSTRACT

In digital predistortion (DPD) implementation, the computational complexity of coefficients estimation of the DPD model is a key performance metric. Conventional coefficients estimation algorithms, such as least squares (LS), recursive least squares (RLS), and least mean squares (LMS) cannot achieve a fast convergence with little computation. In this paper, we propose an RLS algorithm with reduced complexity by introducing orthonormal polynomial basis functions. The proposed algorithm is as simple as LMS algorithm yet as efficient as RLS algorithm. Simulation results validate our analysis.

***Index Terms*—** Nonlinearity, predistorter, orthonormal, recursive least squares, reduced complexity

1. INTRODUCTION

In modern communication systems, power amplifiers (PAs) are always driven into nonlinear region to improve its efficiency. DPD is a widely used technique to improve the power efficiency of the PA while maintaining its linearity [1].

In terms of the DPD implementation, the computational complexity of coefficients estimation of the DPD model is a key performance metric. For example, the polynomial model is a popular model in adaptive DPD systems as it is intuitive to describe the characteristics of different PAs [2]. LS algorithm can be applied for coefficients estimation. However, it is difficult to update the model coefficients in real-time by inverting the data matrix directly [3]. For real-time processing, RLS and LMS algorithms are attractive because the model coefficients can be updated sample by sample. RLS algorithm replaced the matrix inversion with sample-by-sample vector processing and reduced the computational complexity. LMS algorithm consumes less hardware than the RLS algorithm. However, the convergence speed of the LMS algorithm is slower than that of the RLS algorithm. In this paper, by introducing orthonormal polynomial basis functions, we propose an RLS algorithm with reduced complexity for the polynomial models. The proposed algorithm is as simple as LMS algorithm yet as efficient as RLS algorithm.

The rest of the paper is organized as follows: Section 2 describes the architecture of an adaptive DPD system, and introduces the conventional algorithms for predistorter coefficients estimation. Section 3 defines the notation of the RLS algorithm with reduced complexity, and discusses its hardware complexity comparing with conventional LMS and RLS algorithms. Section 4 shows the benefits of the proposed algorithm in terms of convergence speed and linearization performance by simulation results. Section 5 concludes this paper.

S. Yao is with Shanghai Institute of Microsystem and Information Technology, CAS, and Key Laboratory of Wireless Sensor Network and Communication, CAS. H. Qian and K. Kang are with Shanghai Research Center for Wireless Communications, Shanghai Institute of Microsystem and Information Technology, CAS, and Shanghai Internet of Things Co., Ltd. M. Shen is with Institute of Wireless Communications, Shanghai Jiao Tong University.

2. SYSTEM ARCHITECTURE AND CONVENTIONAL ALGORITHMS

2.1. System Architecture

Adaptive DPD technique is attractive among all PA linearization techniques as they providing good trade-off between linearization performance and implementation cost. In most practical implementations, indirect learning architecture [4] with polynomial model is used to compensate the nonlinearity of the PA. Fig. 1 show an adaptive DPD system with

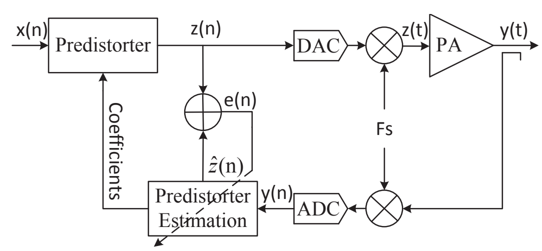

Fig. 1. System architecture of an adaptive DPD system.

indirect learning architecture. As shown in Fig. 1, $x(n)$ is the input of the predistorter, $y(n)$ is the output of the PA and $z(n)$ is the output of the predistorter as well as the input of the PA. The PA output and input, $y(n)$ and $z(n)$, are used to estimate coefficients of the predistorter, where $z(n)$ is obtained from input samples $x(n)$ given DPD coefficients from the previous iteration. After the DPD coefficients estimation, DPD coefficients in the predistorter block will be updated accordingly.

2.2. Conventional Algorithms

For predistorter estimation block in Fig. 1, polynomial model derived in [2] can be applied,

$$\hat{z}(n) = \sum_{k=1}^{K} \alpha_{2k-1}(n)|y(n)|^{2(k-1)}y(n) = \boldsymbol{\phi}(n)\boldsymbol{\alpha}(n). \quad (1)$$

where $\boldsymbol{\phi}(n) = [\phi_1(n), \phi_3(n), \cdots, \phi_{2K-1}(n)]$, $\phi_k(n) = y(n)|y(n)|^{2(k-1)}$, $\boldsymbol{\alpha}(n) = [\alpha_1(n), \alpha_3(n), \cdots, \alpha_{2K-1}(n)]^T$ and $(\cdot)^T$ denotes the matrix transpose. The error signal is presented as $e(n) = z(n) - \hat{z}(n) = z(n) - \boldsymbol{\phi}(n)\boldsymbol{\alpha}(n)$. The objective function of LS algorithm is given by

$$\arg\min_{\boldsymbol{\alpha}} \sum_{n=1}^{N} |e(n)|^2 = \arg\min_{\boldsymbol{\alpha}} \sum_{n=1}^{N} |z(n) - \boldsymbol{\phi}(n)\boldsymbol{\alpha}(n)|^2.$$

The LS solution of $\boldsymbol{\alpha}(n)$ is derived as [5]

$$\boldsymbol{\alpha}(n) = \left(\boldsymbol{\Phi}^H(n)\boldsymbol{\Phi}(n)\right)^{-1}\boldsymbol{\Phi}^H(n)\boldsymbol{z}(n), \quad (2)$$

where $\boldsymbol{z}(n) = [z(1), z(2), \cdots, z(n)]^T$,

$$\boldsymbol{\Phi}(n) = \begin{bmatrix} \boldsymbol{\phi}(1) \\ \boldsymbol{\phi}(2) \\ \vdots \\ \boldsymbol{\phi}(n) \end{bmatrix}, \quad (3)$$

and $(\cdot)^H$ means the Hermitian transpose.

The predistorter coefficients $\boldsymbol{\alpha}(n)$ can not be updated in real-time as the LS algorithm works on a block of data samples. Furthermore, it is difficult to invert a matrix directly on hardware platforms, especially when the matrix is ill-conditioned. For real-time processing, RLS and LMS algorithms are attractive because the model coefficients can be updated sample by sample.

The procedure of the RLS adaptation is given by [6]:

$$\boldsymbol{P}(n) = \left(\boldsymbol{I} - \frac{\boldsymbol{P}(n-1)\boldsymbol{\phi}^H(n)\boldsymbol{\phi}(n)}{1 + \boldsymbol{\phi}(n)\boldsymbol{P}(n-1)\boldsymbol{\phi}^H(n)}\right)\boldsymbol{P}(n-1), \quad (4)$$

$$\boldsymbol{\alpha}(n) = \boldsymbol{\alpha}(n-1) + \frac{\boldsymbol{P}(n-1)\boldsymbol{\phi}^H(n)}{1 + \boldsymbol{\phi}(n)\boldsymbol{P}(n-1)\boldsymbol{\phi}^H(n)} (z(n) - \boldsymbol{\phi}(n)\boldsymbol{\alpha}(n-1)), \quad (5)$$

where $\boldsymbol{P}(n)$ is the inversion of $\boldsymbol{\Phi}^H(n)\boldsymbol{\Phi}(n)$ and is initialized by $\boldsymbol{P}(0) = \lambda^{-1}\boldsymbol{I}$, λ is an arbitrary small constant. In the context of predistorter coefficients estimation, it is natural to set $\boldsymbol{\alpha}(0) = [1, 0, \cdots, 0]^T$.

The procedure of the LMS adaptation is given by [6]:

$$\boldsymbol{\alpha}(n) = \boldsymbol{\alpha}(n-1) + \delta(z(n) - \boldsymbol{\phi}(n)\boldsymbol{\alpha}(n-1))\boldsymbol{\phi}^H(n), \quad (6)$$

where δ is the step-size of LMS algorithm.

3. AN RLS ALGORITHM WITH REDUCED COMPLEXITY

Orthogonal basis functions presented in [5, 7] can help to alleviate the numerical instability during model coefficients estimation. In this paper, we define orthonormal basis functions $\psi_k(n)$ on the fundamental of orthogonal basis functions. For basis functions $\psi_k(n)$ and $\psi_l(n)$, they are orthonormal if the following condition is satisfied:

$$E[\psi_k^*(n)\psi_l(n)] = \frac{1}{N}\sum_{n=1}^{N} \psi_k^*(n)\psi_l(n) = \begin{cases} 0, & \forall k \neq l, \\ 1, & \forall k = l. \end{cases}$$

With orthonormal basis functions, $\boldsymbol{\Phi}(n)$, $\boldsymbol{\phi}(n)$ and $\phi_k(n)$ are corresponding to $\boldsymbol{\Psi}(n)$, $\boldsymbol{\psi}(n)$ and $\psi_k(n)$. The LS solution in Eq. (2) can be rewritten as

$$\boldsymbol{\beta}(n) = \left(\boldsymbol{\Psi}^H(n)\boldsymbol{\Psi}(n)\right)^{-1}\boldsymbol{\Psi}^H(n)\boldsymbol{z}(n). \quad (7)$$

According to the orthonormality, $\boldsymbol{\Psi}^H(n)\boldsymbol{\Psi}(n) = n\boldsymbol{I}$, Eq. (7) is simplified as

$$\boldsymbol{\beta}(n) = \frac{1}{n}\boldsymbol{\Psi}^H(n)\boldsymbol{z}(n). \quad (8)$$

Unfold $\boldsymbol{\Psi}^H(n)\boldsymbol{z}(n)$, we have

$$\boldsymbol{\Psi}^H(n)\boldsymbol{z}(n) = \boldsymbol{\Psi}^H(n-1)\boldsymbol{z}(n-1) + \boldsymbol{\psi}^H(n)z(n). \quad (9)$$

Combining Eq. (8) and (9),

$$\boldsymbol{\beta}(n) = \boldsymbol{\beta}(n-1) + \frac{1}{n}\left(z(n) - \boldsymbol{\psi}(n)\boldsymbol{\beta}(n-1)\right)\boldsymbol{\psi}^H(n) \quad (10)$$

As indicated in Eq. (10), the LS solution $\boldsymbol{\beta}(n)$ is updated recursively, this algorithm should have the almost same performance as RLS algorithm and the hardware complexity is significantly reduced. In practical implementation, $1/n$ can be calculated by a look up table (LUT) with n as its index. The hardware complexity of the proposed algorithm is almost the same as that of the LMS algorithm.

We learn from Eq. (6) and (10) that the only difference is δ and $1/n$. For the LMS algorithm, when the δ is set to a large number, the LMS algorithm converges quickly with a large overshoot; when the δ is set to a small number, the LMS algorithm converges slowly with a small overshoot. It is not easy to choose a suitable δ for the LMS algorithm for different application, on the other words, it is a disadvantage of this algorithm. By replacing δ with $1/n$, the proposed algorithm can converges quickly with a small overshoot.

Be similar to orthogonal basis functions [5, 7], we need to find an upper triangular matrix $\boldsymbol{U}$, which constructs $\boldsymbol{\Psi}(n) = \boldsymbol{\Phi}(n)\boldsymbol{U}$. For the data matrix,

$$\boldsymbol{\Psi}^H(n)\boldsymbol{\Psi}(n) = \boldsymbol{U}^H\boldsymbol{\Phi}^H(n)\boldsymbol{\Phi}(n)\boldsymbol{U} = n\boldsymbol{I}. \quad (11)$$

We learn for Eq. (11) that for a given distribution of $|y|$, $\boldsymbol{\Phi}^H(n)\boldsymbol{\Phi}(n)$ can be obtained, then the corresponding upper triangular matrix $\boldsymbol{U}$ can be solved. For example, $|y|$ is uniformly distributed in $[0,1]$, the (k,l)-th element of $\mathbf{\Phi}^H(n)\mathbf{\Phi}(n)$ is given by

$$\begin{aligned}\sum_{m=1}^{n}|y(m)|^{2k+2l-2} &= nE[|y|^{2k+2l-2}] \\ &= n\int_{|y|}|y|^{2k+2l-2}f(|y|)d|y| \\ &= n\frac{1}{2k+2l-1}.\end{aligned}$$

then the (k,l)-th element of $\boldsymbol{U}$ can be derived as

$$U_{l,k} = \begin{cases} (-1)^{l+k}\frac{(2l+2k-3)!\sqrt{4k-1}}{4^{k-1}(k-l)!(2l-1)!(k+l-2)!}, & l \leq k, \\ 0, & l > k. \end{cases}$$

If $|y|$ follows other distributions, the (k,l)-th element of data matrix $\mathbf{\Phi}^H(n)\mathbf{\Phi}(n)$ can also be solved. The closed-form expression of the upper triangular matrix $\boldsymbol{U}$ may not be available. However, the corresponding $U_{l,k}$ can still be solved iteratively by setting the dimension of the matrix $\mathbf{\Phi}^H(n)\mathbf{\Phi}(n)$ in Eq. (11) to $1,2,\ldots$.

4. SIMULATION PERFORMANCE

4.1. Convergence Speed

In order to verify the convergence speed of three different algorithms, we apply an ARCTAN PA model

$$y = (\gamma_1 tan^{-1}(\zeta_1|z|) + \gamma_2 tan^{-1}(\zeta_2|z|))e^{j\angle z}, \quad (12)$$

where $\gamma_1 = 8.00335 - j4.61157$, $\gamma_2 = -3.77167 + j12.03758$, $\zeta_1 = 2.26895$ and $\zeta_2 = 0.8234$ [5]. The input is a random signal distributed in $[0,1]$ and the SNR is 30dB. The orthonormal basis functions with $K = 9$ is applied.

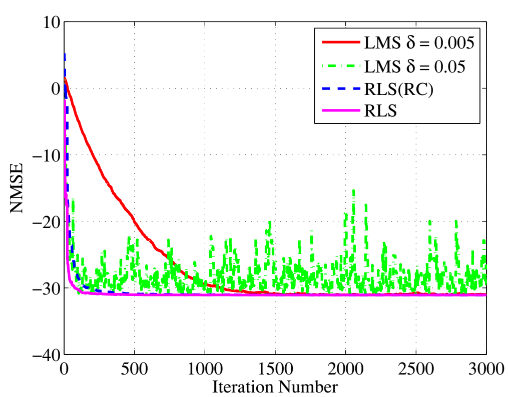

Fig. 2. Convergence speed of three different algorithms.

Fig. 2 shows the normalized mean squares error (NMSE) vs. iteration number for three different algorithms. From left to right, the pink line shows the convergence speed for RLS algorithm, the blue line shows the convergence speed for the proposed algorithm and the green line and red line show the convergence speed for LMS algorithms with $\delta = 0.05$ and $\delta = 0.005$ respectively.

As indicated in Fig. 2, the proposed algorithm achieves almost the same performance with RLS algorithm; the two lines of LMS algorithm reflect the trade-off between convergence speed and overshoot.

Table 1 shows the differences among the different simulation case in Fig. 2. We learn from Table 1 that the proposed algorithm performs the almost same performance with RLS algorithm with the same hardware complexity with LMS algorithm. When $\delta = 0.05$, the LMS algorithm conveges quickly but the overshot is 15dB; when $\delta = 0.005$, the LMS algorithm conveges slowly but the overshoot is 0.5dB.

Table 1. The differences among three algorithms.

Algorithm	Multipliers	Conv-Speed	Instability
RLS	60	200	0dB
RLS(RC)	11	300	0dB
LMS(0.05)	11	300	15dB
LMS(0.005)	11	1500	0.5dB

The orthonormal basis functions are derived by assuming that the input signal is uniformly distributed and the polynomial model is memoryless. If the input signals are with other distributions or the polynomial model is memory, the data matrix $\mathbf{\Phi}^H(n)\mathbf{\Phi}(n)$ is not an identity matrix, but the proposed algorithm works as the numerical instability is still alleviated. For example, the input signal is a 20MHz bandwidth IEEE802.11g signal, the same trends of convergence speed for three different algorithms are shown in Fig. 3.

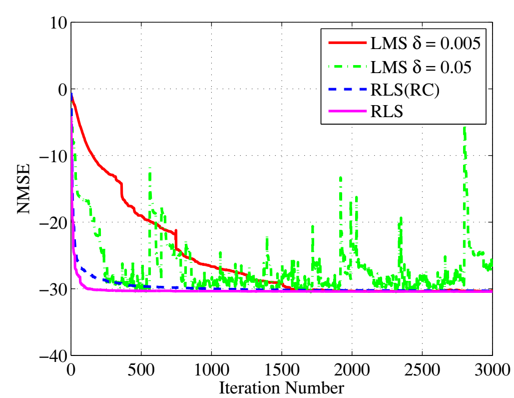

Fig. 3. Convergence speed of three different algorithms.

4.2. Linearization Performance

To compare the linearization performance of the three algorithms, we test a 20MHz bandwidth IEEE802.11g signal. In general, orthogonal frequency division multiplexing (OFDM) signals are regarded as complex gaussian distributed signals. We use the same ARCTAN PA model in 4.1 and the orthonormal basis functions with $K = 9$.

In our simulation, we drive the PA to the nonlinear region, and test the performance for the three algorithms. The power spectral destiny (PSD) at the PA output are shown in Fig. 4. In Fig. 4, from top to bottom, the red line shows the PSD

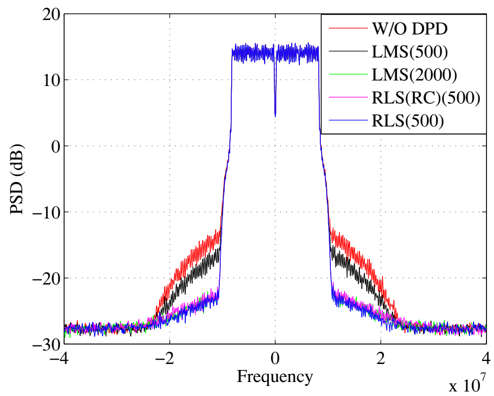

Fig. 4. PA output PSD.

at the PA output without DPD, the black line shows the PSD at the PA output with LMS algorithm ($\delta = 0.005$) using 500 samples, the green line shows the PSD at the PA output with LMS algorithm using 2000 samples, the pink line shows the PSD at the PA output with the proposed algorithm using 500 samples, the blue line shows the PSD at the PA output with RLS algorithm using 500 samples.

As indicated in Fig. 4, with the same iteration number 500, the the proposed algorithm and RLS algorithm are converged but the LMS algorithm is not converged; with the iteration number 2000, the LMS algorithm achieves the almost same performance with the proposed and the RLS algorithm.

5. CONCLUSION

In most predistorter designs, the polynomial model is a popular model in adaptive DPD systems as it is intuitive to describe the characteristics of different PAs. The computational complexity of coefficients estimation of the DPD model is a key performance metric. LS algorithm cannot update the model coefficients in real-time as it works on a block of data samples. RLS and LMS algorithms are attractive because the model coefficients can be updated sample by sample. RLS algorithm replaced the matrix inversion with sample-by-sample vector processing and reduced the computational complexity. LMS algorithm consumes less hardware than the RLS algorithm. However, the convergence speed of the LMS algorithm is slower than that of the RLS algorithm. In this paper, by introducing orthonormal polynomial basis functions, we propose an RLS algorithm with reduced complexity for the polynomial models. In comparison, we test the three algorithms in terms of convergence speed and linearization performance. Simulation results show that the proposed algorithm is as simple as LMS algorithm yet as efficient as RLS algorithm.

6. ACKNOWLEDGMENT

This work was supported in part by the 100 Talents Program of Chinese Academy of Sciences, the Shanghai Pujiang Talent Program (No. 11PJ1408700), International Science and Technology Cooperation project of Shanghai (No.11220705400), National High Technology Research and Development Program ("863"Program) of China (No. 2011AA01A109), and Shanghai Municipal Commission of Economy and Information(No. YORSKD1001).

7. REFERENCES

[1] P. L. Gilabert, A. Cesari, G. Montoro, E. Bertran, and J. M. Dilhac, "Multi-lookup table fpga implementation of an adaptive digital predistorter for linearizing rf power amplifiers with memory effects," *IEEE Transactions on Microwave Theory and Techniques*, vol. 56, pp. 372–384, February 2008.

[2] L. Ding, G. T. Zhou, D. R. Morgan, Z. X. Ma, J. S. Kenney, J. Kim, and C. R. Giardina, "A robust digital baseband predistorter constructed using memory polynomials," *IEEE Transactions on Communications*, vol. 52, pp. 159–165, January 2004.

[3] P. Salmela, A. Happonen, A. Burian, and J. Takala, "Several approaches to fixed-point implementation of matrix inversion," in *International Symposium on Signals, Circuits and Systems*, July 2005, pp. 497 – 500.

[4] C. Eun and E.J Powers, "A new volterra predistorter based on the indirect learning architecture," *IEEE Transactions on Signal Processing*, vol. 45, pp. 223–227, January 1997.

[5] R. Raich, H. Qian, and G. T. Zhou, "Orthogonal polynomials for power amplifier modeling and predistorter design," *IEEE Transactions on Vehicular Technology*, vol. 53, pp. 1468–1479, September 2004.

[6] V. J. Mathews, "Adaptive polynomial filters," *IEEE Signal Processing Magazine*, vol. 8, pp. 10 – 26, July 1991.

[7] R. Raich and G. T. Zhou, "Orthogonal polynomials for complex gaussian processes," *IEEE Transactions on Signal Processing*, vol. 52, pp. 2788–2797, October 2004.