A Method for Selecting Online the Coefficients to be Updated in a DPD for PA Linearization

Quynh Anh Pham[(1)], Gabriel Montoro[(1)], David López-Bueno[(2)(1)], Pere L. Gilabert[(1)]
thi.quynh.anh.pham@upc.edu, gabriel.montoro@upc.edu, david.lopez@cttc.es, plgilabert@tsc.upc.edu
[(1)]Universitat Politècnica de Catalunya
[(2)]Centre Tecnològic de Telecomunicacions de Catalunya

***Abstract*—This paper presents a technique for selecting online the coefficients to be updated in a digital predistorter (DPD) based on direct learning. The proposed method, which is based on a combination of matching pursuit (MP) and least squares (LS) techniques (and is therefore named MP-LS method) allows to improve the power amplifier (PA) linearization performance of a fixed number of DPD coefficients, due to the fact that at each DPD iteration the coefficients to be updated are properly chosen. The proposed technique is compared to a conventional LS estimation, and experimental results demonstrate that the MP-LS method can provide a performance improvement in relation to a DPD with fixed-preselected coefficients. The method could be especially useful in DPD systems that have hardware restrictions in the resources to be used by the update subsystem in the feedback path. That is the case of DPDs based on FPGA devices implementing a QR algorithm in the programmable logic (PL) side.**

***Index Terms*—Digital predistortion, matching pursuit, power amplifier linearization.**

I. Introduction

Digital predistortion (DPD) [1] becomes a common strategy to deal with the trade-off between linearity and efficiency of power amplifiers in wireless communication systems. In new generation of wireless communication systems, such as 4G and 5G, where the bandwidth and the peak-to-average power ratios (PAPR) of the signals is ever-growing, the number of parameters of the DPD linearizers has to be increased to eliminate the PAs' unwanted nonlinear distortion effects. This leads not only to the upgrowth of the computational complexity, but also to the uncertainty of the LS estimation in DPD linearization.

Recently, many efforts have been made to decrease the number of basis functions of the DPD linearizers to avoid the uncertainty and overfitting of the LS calculation [2], [3], [4]. Among the proposed techniques, orthogonal matching pursuit (OMP) [2] is a popularly used solution for DPD order reduction. Thus, OMP is usually calculated offline to select the most relevant basis functions for the DPD linearizer in the forward path. However, due to the high computational cost of OMP, it becomes inefficient to be implemented online in the DPD feedback path to reduce the number of DPD parameters.

Therefore, this paper presents a proposed approach that combines matching pursuit (MP) and LS to extract the DPD parameters in the feedback path. Matching pursuit, that has lower computational cost, is an ideal solution for selecting the most important basis functions of DPD feedback path. It enables to reduce the number of DPD parameters and avoid the overfitting and uncertainty of LS estimation.

II. Overview on Matching Pursuit Techniques

A. Matching Pursuit and Orthogonal Matching Pursuit

The problem is stated as follows: Given a signal $\mathbf{y}$ and a dictionary $\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_M\}$ where $\mathbf{u}_m$, $m = 1, 2, ..., M$, are normalized functions (i.e. $||\mathbf{u}_m||_2 = 1$), find a linear expansion of the signal $\mathbf{y}$ in terms of functions $\mathbf{u}_m$

$$\mathbf{y} = \sum_{m=1}^{M} a_m \mathbf{u}_m \tag{1}$$

where a_m is the scalar weighting factor for the function $\mathbf{u}_m$. In order to solve the problem (1), the basic idea is to find an approximate representation of $\mathbf{y}$ as a sum of functions $\mathbf{u}_k$ as following

$$\mathbf{y} \approx \hat{\mathbf{y}} = \sum_{k=1}^{K} a_k \mathbf{u}_k \tag{2}$$

It is expected that the residual of the signal $\mathbf{y}$ and its approximation in (2) is minimum. That is

$$\mathbf{r} = ||\mathbf{y} - \hat{\mathbf{y}}||_2 = ||\mathbf{y} - \sum_{k=1}^{K} a_k \mathbf{u}_k||_2 = \min \tag{3}$$

The criterion (3) is a combinatorial-explosion problem since it requires to examine all the possible K-function combinations (i.e. subsets of K functions) from the dictionary $\mathbf{U}$ that includes M functions.

Matching pursuit, firstly proposed by Mallat and Zhang in [5], is an iterative procedure finding the sub-optimal solution for (3). At each iteration, the selected element is determined based on the inner product between the current approximation error $\mathbf{r}$ and the functions (i.e. columns) in $\mathbf{U}$. The matching pursuit algorithm is summarized in Algorithm 1.

On the other hand, orthogonal matching pursuit (OMP) differs from MP in the way how the residual error is evaluated at each iteration, being calculated the residual error orthogonal to the subspace created by the selected basis. OMP is computationally more expensive than MP but provides a better basis sorting.

Algorithm 1 Matching Pursuit

```
procedure MP(U, y, K)
    initialization:
    r^(0) = y;   S(0) = {};   n = 0;
    for n = 1 to K do
        i^(n) ← argmax_i |U^H_{i} r^(n-1)|;
        S(n) ← S(n-1) ⋃ {i^(n)};
        a^n ← U^H_{i} r^(n-1);
        r^(n) ← r^(n-1) - a^n U_{i};
    end for
    Return S
end procedure
```

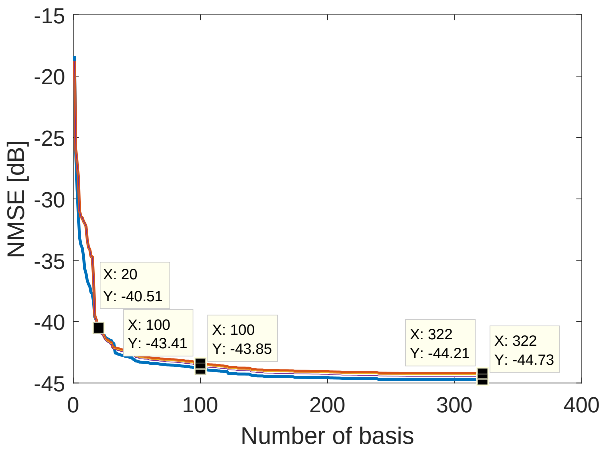

Fig. 1. NMSE of a PA modeling in relation to the number of selected OMP basis (the training has been done offline by only using the signal in blue).

B. Orthogonal Matching Pursuit and Sorting Robustness

Fig. 1 shows the evolution of the NMSE of a PA identification procedure when the OMP sorting is obtained from an initial training signal and later is evaluated by using another set of signals, different from the one used for training but of the same characteristics. These experimental results have been obtained by using the testbed described in Section IV. According to Fig. 1, there is a not significant degradation in the performance of the OMP selected basis. Therefore, we can consider the offline training procedure good enough for selecting the necessary OMP basis to be used for the DPD system.

III. Structure of the Proposed DPD

First, in the DPD forward path, the number of coefficients is reduced by applying offline OMP to choose the most relevant basis functions of $\mathbf{U}$. Second, in the feedback path, one of the most popular solution is to extract the best coefficients, according to a LS solution, by means of the QR algorithm.

A. Forward Path

The input-output relationship of the DPD block in the forward path of the DPD system (see Fig. 1) is as follows

$$\mathbf{x} = \mathbf{u} - \mathbf{U}\mathbf{w} \tag{4}$$

in which, $\mathbf{u} = (u[0], \cdots, u[n], \cdots, u[N-1])^T$, where $n = 0, \cdots, N-1$, is the $N \times 1$ input signal; $\mathbf{x} = (x[0], \cdots, x[n], \cdots, x[N-1])^T$ is the $N \times 1$ predistorted signal; $\mathbf{w} = \Big(w_1[n], \cdots, w_i[n], \cdots, w_M[n]\Big)^T$ is the vector of coefficients at time n with dimensions $M \times 1$, and with M being the number of original basis functions describing a particular behavioral model. The $N \times M$ data matrix $\mathbf{U}$ is defined as

$$\mathbf{U} = (\boldsymbol{\varphi_u}[0], \cdots, \boldsymbol{\varphi_u}[n], \cdots, \boldsymbol{\varphi_u}[N-1])^T \tag{5}$$

where $\boldsymbol{\varphi_u}^T[n] = \Big(\phi^u_1[n], \cdots, \phi^u_i[n], \cdots, \phi^u_M[n]\Big)$ is the vector containing the specific basis functions $\phi^u_i[n]$ (with $i = 1, \cdots, M$) at time n. The general definition in (5) can be particularized for any DPD behavioral model. In our paper, the generalized memory polynomial (GMP) behavioral model [6] is used to generate $\mathbf{U}$ for linearization purposes. The OMP algorithm is applied to choose the most important basis functions from the original set of basis functions generated by the GMP.

B. Online DPD Estimation/Adaptation with MP and LS in Feedback Path

In a DPD feedback path subsystem (see Fig. 2), the iterative update of the DPD coefficients is expressed as follows

$$\mathbf{w}_{i+1} = \quad \mathbf{w}_i + \mathbf{dw} \tag{6}$$

with $\mathbf{w}_i$ being the $M \times 1$ vector of coefficients of the DPD model at the i^{th} iteration. The coefficients' increment $\mathbf{dw}$ is commonly estimated solving LS

$$\mathbf{dw} = \quad \mu(\mathbf{U}^H\mathbf{U})^{-1}\mathbf{U}^H\mathbf{e} \tag{7}$$

where μ is the learning-rate parameter. The linearization error is defined as $\mathbf{e} = \dfrac{\mathbf{y}}{G_0} - \mathbf{u}$, where G_0 is the desired PA linear gain, the $N \times 1$ vectors $\mathbf{y}$ and $\mathbf{u}$ are system input signal (i.e. before DPD) and the PA output signal, respectively; and the $N \times M$ data matrix $\mathbf{U}$ contains the M basis functions describing the DPD behavioral model.

In this paper, an alternative to the conventional LS method is proposed, consisting in a combination of online matching pursuit and LS (MP-LS) in the feedback path. In a first step, the matching pursuit algorithm (Algorithm 1) is implemented online inside the DPD adaptation loop to choose only the most relevant basis functions to be used for calculating the DPD coefficients. Then, in the second step, the DPD coefficients of feedback path are extracted solving LS.

Fig. 2 represents a conventional direct learning DPD, and Fig. 3 visualizes the proposed DPD estimation/adaptation employing MP-LS approach. Thanks to the MP algorithm, the number of basis functions at each iteration of the DPD

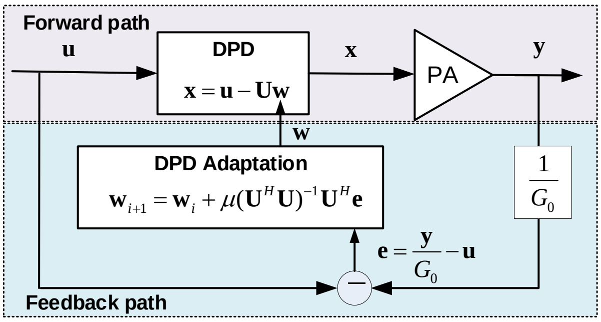

Fig. 2. Adaptive DPD system following a direct learning approach.

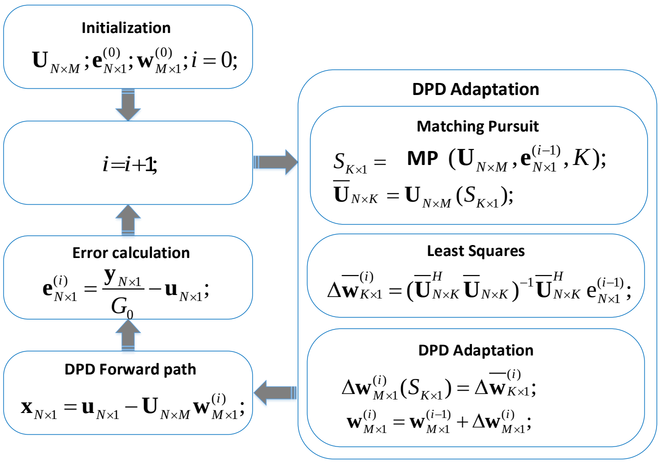

Fig. 3. Flowchart of the DPD estimation/adaptation using the MP-LS technique.

adaptation loop is reduced from M to K, with $K \ll M$. Therefore, the advantage of the MP-LS in relation to the conventional LS approach is this dimensionality reduction in the basis used for coefficients update.

C. Computational Cost and Implementation Issues

The order of magnitude of the computational complexity of the MP algorithm is $O(NM)$ per iteration [7] whereas the computational cost of QR decomposition (necessary for LS solving) is $O(NM^2)$ per iteration [8]. Therefore, the proposed approach helps to reduce the computational cost of the DPD adaptation (usually based on a QR algorithm) but adding an additional cost due to the MP procedure. By means of Matlab's tic-toc we have measured that for 20 coefficients the time for a QR procedure (programmed in plain Matlab code, and applied to the signals described in Section IV) is around 30 times the time of running MP for selecting 20 basis from 100.

IV. Experimental Results

A Matlab controlled hardware testbed, as shown in Fig. 4, is used to validate the proposed DPD strategy MP-LS. The testbed uses an 80 MHz bandwidth carrier-aggregated fast convolution filter bank multi-carrier (FC-FBCM) signal with subcarrier group deactivation, up-converted to the 875 MHz RF frequency to feed a class-J PA.

TABLE I
Summary of the Performed Tests.

Test	**Forward path** (Alg., No. of coeff.)	**Feedback path** (Alg., No. of coeff.)
a)	OMP, 20 coeff.	LS, 20 coeff.
b)	OMP, 100 coeff.	LS, 100 coeff.
c)	OMP, 100 coeff.	MP-LS, 20 coeff.

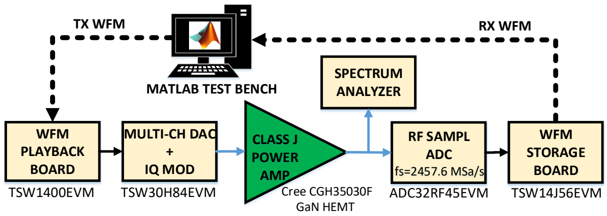

Fig. 4. High-level block diagram of the Matlab-controlled digital linearization test bench.

The original basis functions with 322 coefficients were generated by applying the GMP behavioral model. Then, in the DPD forward path, by means of OMP, the number of coefficients is cut down to 20 or 100, as is explained below. This basis will be later used in the feedback path.

In order to examine the efficiency of the proposed method MP-LS, we executed three tests as following (the tests are summarized in Table I):

Test a) In the forward path, by means of OMP, the number of coefficients is cut down to 20, then, the DPD estimation/adaptation in the feedback path is performed by solving LS by using the Matlab's mldivide or backslash function (which is equivalent to a QR decomposition) using 20 coefficients;

Test b) In the forward path, by means of OMP, the number of coefficients is cut down to 100, and the DPD estimation/adaptation is operated by LS (Matlab's mldivide function) using 100 coefficients;

Test c) In the forward path, the number of coefficients is cut down to 100 using OMP, then, in the feedback path, the basis functions is further reduced to 20 by applying MP, and then the DPD estimation/adaptation is executed by means of LS.

As can be seen in Fig. 5 and Fig. 6, after only 4 or 5 iterations, all DPD adaptation methods have become converged. However, the linearity performance (in terms of NMSE and ACPR) of the *test b*) is the best among three, while the *test a*) is the worst one. Our proposed method MP-LS that enables to reduce the number of necessary coefficients for DPD adaptation/estimation (to 20 coefficients, one fifth to the number of coefficients of *test b*)) gives better performance than the *test a*) and only 2dB less than the performance of *test b*).

Finally, Fig. 7 shows the unlinearized and linearized spectra considering the proposed selection and update of the DPD coefficients.

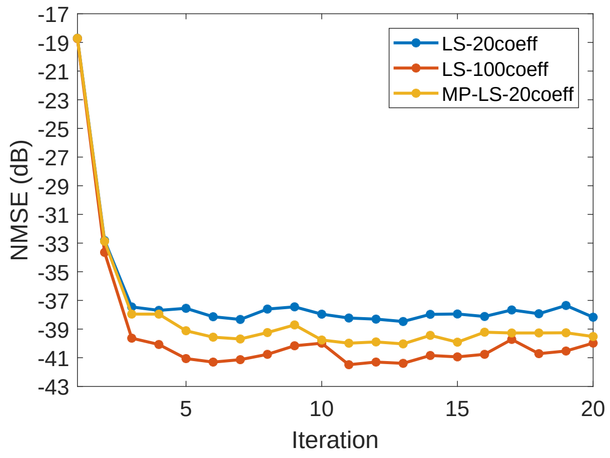

Fig. 5. Evolution of the NMSE for: LS with 20 coefficients, LS with 100 coefficients, and MP-LS with 20 coefficients.

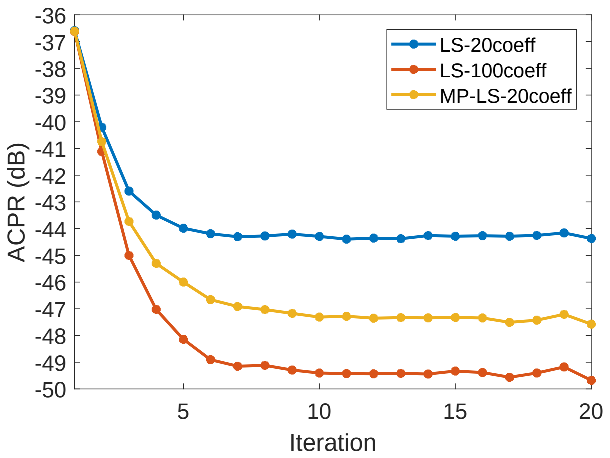

Fig. 6. Evolution of the ACPR for: LS with 20 coefficients, LS with 100 coefficients, and MP-LS with 20 coefficients.

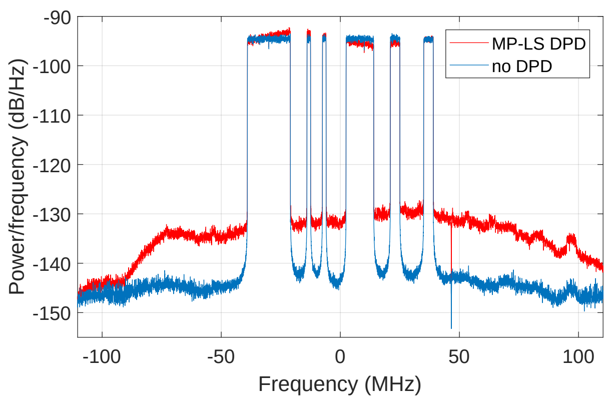

Fig. 7. Spectra of the PA output before and after MP-LS DPD linearization.

V. Conclusion

In this paper, a method that combines matching pursuit (MP) and least squares (LS) to properly choose the amount of coefficients to be estimated in a direct learning DPD has been proposed and tested. The MP-LS method can be used *i*) for improving the performance (ACPR, NMSE) of a DPD system with fixed and bounded amount of coefficients (due to hardware restrictions), or *ii*) for reducing the amount of DPD coefficients to be updated but minimizing the impact in performance degradation.

Acknowledgment

This work was partially supported by the Spanish Government (MICINN) and FEDER under project TEC2017-83343-C4-2-R, and by the Generalitat de Catalunya under grants 2017 SGR 891 and 813.

References

[1] R. N. Braithwaite, *Digital Processing for Front End in Wireless Communication and Broadcasting*. Cambridge University Press, 2011, ch. General principles and design overview of digital predistortion, pp. 143–191.

[2] J. Reina-Tosina, M. Allegue et al., "Behavioral modeling and predistortion of power amplifiers under sparsity hypothesis," *IEEE Trans. on Microw. Theory and Tech.*, vol. 63, no. 2, pp. 745–753, Feb. 2015.

[3] P. L. Gilabert, G. Montoro, D. Lopez, N. Bartzoudis, E. Bertran, M. Payaro, and A. Hourtane, "Order reduction of wideband digital predistorters using principal component analysis," in *Microwave Symposium Digest (IMS), 2013 IEEE MTT-S International*, 2013, pp. 1–4.

[4] Q. A. Pham, D. López-Bueno, T. Wang, G. Montoro, and P. L. Gilabert, "Multi-dimensional LUT-based digital predistorter for concurrent dual-band envelope tracking power amplifier linearization," in *Proc. 2018 IEEE Topical Conf. on RF/Microw. Power Amplifiers for Radio and Wireless Appl. (PAWR)*, Jan. 2018, pp. 47–50.

[5] S. Mallat and Z. Zhang, "Adaptive time-frequency decomposition with matching pursuits," in *[1992] Proceedings of the IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*. IEEE.

[6] D. R. Morgan, Z. Ma et al., "A Generalized Memory Polynomial Model for Digital Predistortion of RF Power Amplifiers," *IEEE Trans. on Signal Processing*, vol. 54, no. 10, pp. 3852–3860, Oct. 2006.

[7] T. Blumensath and M. Davies, "Gradient pursuits," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2370–2382, jun 2008.

[8] F. Westad, K. Diepold, and H. Martens, "Qr-plsr: Reduced-rank regression for high-speed hardware implementation," *Journal Of Chemometrics*, vol. 10, pp. 439–451, 1996.