Dynamic Selection and Estimation of the Digital Predistorter Parameters for Power Amplifier Linearization

Quynh Anh Pham, *Student Member, IEEE,* Gabriel Montoro, *Member, IEEE,* David López–Bueno, *Senior Member, IEEE,* and Pere L. Gilabert, *Senior Member, IEEE*

***Abstract*—This paper presents a new technique that dynamically estimates and updates the coefficients of a digital predistorter (DPD) for power amplifier (PA) linearization. The proposed technique is dynamic in the sense of estimating, at every iteration of the coefficient's update, only the minimum necessary parameters according to a criterion based on the residual estimation error. At the first step, the original basis functions defining the DPD in the forward path are orthonormalized for DPD adaptation in the feedback path by means of a precalculated principal components analysis (PCA) transformation. The robustness and reliability of the precalculated PCA transformation (i.e., PCA transformation matrix obtained off-line and only once) is tested and verified. Then, at the second step, a properly modified partial least squares (PLS) method, named dynamic partial least squares (DPLS), is applied to obtain the minimum and most relevant transformed components required for updating the coefficients of the DPD linearizer. The combination of the PCA transformation with the DPLS extraction of components is equivalent to a canonical correlation analysis (CCA) updating solution, which is optimum in the sense of generating components with maximum correlation (instead of maximum covariance as in the case of the DPLS extraction alone). The proposed dynamic extraction technique is evaluated and compared in terms of computational cost and performance with the commonly used QR decomposition approach for solving the least squares (LS) problem. Experimental results show that the proposed method (i.e., combining PCA with DPLS) drastically reduces the amount of DPD coefficients to be estimated while maintaining the same linearization performance.**

***Index Terms*—Canonical correlation analysis, digital predistortion, model order reduction, partial least squares, power amplifier, principal component analysis.**

I. Introduction

Digital predistortion (DPD) linearization can overcome or at least mitigate the efficiency versus linearity trade-off in power amplifiers (PAs). In order to avoid wasting excessive power resources when handling high peak-to-average power ratio (PAPR) signals, high efficient amplification architectures based on dynamic load or dynamic supply modulation have been adopted. Some of the most popular solutions proposed in literature (and also adopted by the industry in some cases) include: envelope tracking PAs [1], Doherty PAs [2], load modulated balanced amplifiers (LMBA) [3] and LINC or outphasing PAs [4]. In either case, these highly efficient topologies require the use of linearization techniques to guarantee the linearity levels specified in the communications standards.

Unlike in macro base stations, where the DC power consumption in the transmitter chain is dominated by the PA, when targeting the linearization of small cells (or even handsets) the computational complexity and DC consumption of the digital signal processing and data conversion stages are critical. In addition, to linearize highly efficient amplification architectures based on dynamic supply or dynamic load modulation, the DPD behavioral model requires a huge number of parameters [5]. This can cause an ill-conditioned least squares (LS) estimation and at the same time increases the computational complexity of the overall DPD system. Consequently, a lot of effort has been made to reduce the number of parameters of the DPD [6]–[9].

As depicted in Fig. 1, the DPD linearization system can be divided into two subsystems: a forward path subsystem operating in real-time, where the input signal is conveniently predistorted; and a feedback or observation path subsystem, where the coefficients characterizing the nonlinear DPD function in the forward path are estimated and updated in a more relaxed time scale. When targeting an implementation in a digital signal processing platform, for example, a system-on-chip (SoC) FPGA device, the DPD function in the forward path can be implemented in a programmable logic (PL) unit. For example, by following a LUT approach as in [10], [11], or by considering a polynomial approach using the Horner's rule as in [12], or by combining both complex multipliers/adders and memory as in [13]. Therefore, the DPD function in the forward path should be designed as simple as possible (i.e., including the minimum and most relevant basis functions) to save as many hardware logic resources and memory as possible. On the other hand, the adaptation of the DPD coefficients can be carried out in a processing system (PS) in a much slower time scale than in the forward path (i.e., not in real time).

Dimensionality reduction techniques can be classified into

This paper is an expanded version from the IEEE MTT-S Radio and Wireless Week (PAWR 2019), Orlando, FL, USA, January 20-23, 2019.

This work was supported by the Spanish Government (MINECO) and FEDER under project TEC2017-83343-C4-2-R and RTI2018-099841-B-I00, and by the Generalitat de Catalunya under grant 2017 SGR 813 and 2017 SGR 891.

Q. A. Pham, G. Montoro and P. L. Gilabert are with the Department of Signal Theory and Communications., Universitat Politècnica de Catalunya (UPC) - Barcelona Tech, Barcelona, 08034, SPAIN, e-mail: thi.quynh.anh.pham@upc.edu

D. López–Bueno is with UPC and Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA).

Manuscript received March 12, 2019; revised XXXX XX, 2019.

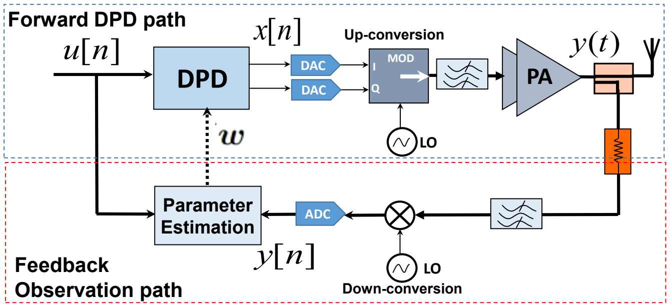

Fig. 1. Block diagram of a closed-loop DPD system following a direct learning approach.

two main groups: a) feature selection techniques such as, LASSO [14], Ridge regression [15], the sparse Bayesian learning algorithm [16] or the orthogonal matching pursuit (OMP) [17], that select the most relevant variables from a random set of original variables; and b) feature extraction techniques, such as, principal component analysis (PCA) [6], partial least squares (PLS) [18] or canonical correlation analysis (CCA) [19], that create a reduced set of new variables that are linear or nonlinear combinations of the original variables.

Some of the aforementioned feature extraction techniques have been proposed by the authors in [20]–[22] as an alternative to one of the most commonly used solutions based on QR factorization combined with recursive least squares (QR-RLS) [23]. The objective of using these techniques is not only to ensure a proper well-conditioned estimation, but also a reduction in the number of parameters required in the identification or adaptation process. However, unlike with feature selection techniques, with feature extraction techniques, the number of coefficients of the DPD function in the forward path is not reduced. For that reason, in [22] the authors proposed the combination of an off-line OMP search for reducing the DPD basis in the forward path with PLS extraction in the adaptation (or observation) path. In comparison to PCA [20], where the new basis of components with maximal variance are obtained taking into account only the input data, with PLS the new basis of components show maximal covariance in relation to the signal to be estimated and thus, the reduction capabilities using PLS are significantly better than using PCA, as discussed in [22]. In [24], the authors extended the reduction capabilities by presenting a dynamic adaptation approach based on PLS where the basis of new components used in the DPD estimation is dynamically adjusted and thus, at every iteration of the adaptation process, only the minimum number of required components are used to minimize the linearization error.

In this paper, the authors go a step further by proposing a new approach to dynamically estimate and update the coefficients of the DPD that not only shows better dimensionality reduction capabilities than previous published solutions in [20], [22], [24], but also introduces less computational complexity than QR-LS (i.e., any method based on a QR decomposition with a fixed number of coefficients used to solve the LS problem, such as Matlab's mldivide or backslash) thanks to the inherent dimensionality reduction introduced by the proposed algorithm.

The proposed DPD adaptation technique is based on the combination of PCA with dynamic PLS (PCA-DPLS), which is equivalent to the CCA, but with significantly lower computational cost. Like in CCA, the proposed PCA-DPLS technique creates a new basis with maximal correlation in relation to the signal to be estimated, which makes it more suitable for reducing the number of components. Therefore, in the PCA-DPLS technique, first, the original basis functions defining the DPD in the forward path are orthonormalized by means of a precalculated PCA transformation matrix (obtained off-line and only once). Second, a modified dynamic PLS method is applied to obtain the minimum and most relevant transformed components required for updating the coefficients of the DPD linearizer.

The remainder of this paper is organized as follows. Section II presents an overview on the feature extraction techniques required to describe the proposed dynamic DPD adaptation approach. Section III describes in detail the proposed joint PCA-DPLS approach. Section IV describes the experimental test bench and shows experimental results proving the advantages of the PCA-DPLS technique to dynamically update the coefficients of the DPD. Finally, the conclusion is given in Section V.

II. Dimensionality Reduction Based on Linear Transformations

A. Forward Path DPD

Following the same notation shown in the block diagram in Fig. 1, in the forward path, the input-output relationship at the DPD block can be described in a matrix notation as

$$\boldsymbol{x} = \boldsymbol{u} - \boldsymbol{U}\boldsymbol{w} \tag{1}$$

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

$$\boldsymbol{U} = (\boldsymbol{\varphi_u}[0], \cdots, \boldsymbol{\varphi_u}[n], \cdots, \boldsymbol{\varphi_u}[N-1])^T \tag{2}$$

where $\boldsymbol{\varphi_u}^T[n] = \Big(\phi_1^u[n], \cdots, \phi_i^u[n], \cdots, \phi_M^u[n]\Big)$ is the vector containing the specific basis functions $\phi_i^u[n]$ (with $i = 1, \cdots, M$) at time n. The general definition in (2) can be particularized for any DPD behavioral model. In this paper, we have considered the generalized memory polynomial (GMP) behavioral model proposed in [25], for linearization purposes. The number of parameters of the GMP model will be discussed in Section IV.

B. Principal Component Analysis

The PCA theory is used to generate a new basis set of orthogonal components that are the linear combinations of the original basis functions contained in the $N \times M$ matrix $\boldsymbol{U}$ through the $M \times M$ transformation matrix $\boldsymbol{R}_{PCA}$ that contains the eigenvectors of the covariance matrix of $\boldsymbol{U}$ (i.e., $cov(\boldsymbol{U}) \approx \boldsymbol{U}^H\boldsymbol{U}$),

$$\tilde{\boldsymbol{U}} = \boldsymbol{U}\boldsymbol{R}_{PCA} \tag{3}$$

The $N \times M$ transformed matrix $\tilde{\boldsymbol{U}}$ contains the principal components (i.e., linear combinations of the original basis functions) oriented to capture the maximum variance in the data. Since the components in the new transformed basis $\tilde{\boldsymbol{U}}$ are orthogonal among them and are sorted according to their relevance (i.e., more relevant components show higher eigenvalue), it is be possible to apply some model order reduction by simply removing the components with smaller eigenvalue [6].

The computational cost of extracting the PCA transformation matrix is not negligible, as it will be further discussed in subsection III-B. However, since it only relies on the statistical properties of the input data, it is possible to pre-calculate it off-line, only once, and then use it for linearization purposes considering different input data sets without loss of generality, as long as the input data statistics do not change, or the signal power levels are not altered significantly. In order to validate the reliability and robustness of using a precalculated PCA transformation matrix, we performed a DPD linearization considering a transformation matrix $\boldsymbol{R}_{PCA}$ calculated taking into account a specific set of data. Fig. 2 shows the linearization performance in terms of normalized mean square error (NMSE) and adjacent channel power ratio (ACPR), when applying a precalculated PCA transformation matrix $\boldsymbol{R}_{PCA}$ to generate the transformed matrix $\tilde{\boldsymbol{U}}$ of orthogonal components considering several input data sets, all different (same modulation and bandwidth, but different PAPR) from the one used for obtaining the transformation matrix.

Therefore, after checking that there was no loss in linearization performance, it is fair to say that, as long as the characteristics of the transmitted signal do not change, the PCA precalculated transformation matrix is reliable and robust enough to be used for DPD purposes. It is robust against the different PAPR values produced when random data is modulated for a given modulation and signal bandwidth configuration. However, the recalculation of the PCA transformation matrix is necessary when the characteristics of the transmitted signal, mainly in terms of bandwidth or power operation conditions, change significantly. When this happens, the required basis functions in matrix $\boldsymbol{U}$ that characterize the PA nonlinear and dynamic behavior may change, and consequently, the PCA transformation matrix should be recalculated. As an alternative to the off-line calculation of the PCA transformation matrix, the adaptive PCA (APCA) method could be included as part of the online (not real-time) processing, as proposed in [21].

C. Canonical Correlation Analysis and Partial Least Squares

CCA and PLS [26] are well-known techniques for feature extraction from a set of variables or basis functions. Covariance and correlation are two different statistical measures for quantifying how the variables are related. The main difference between the CCA and the PLS techniques is that CCA creates new components that maximize the correlation factor ρ_{CCA}, defined as

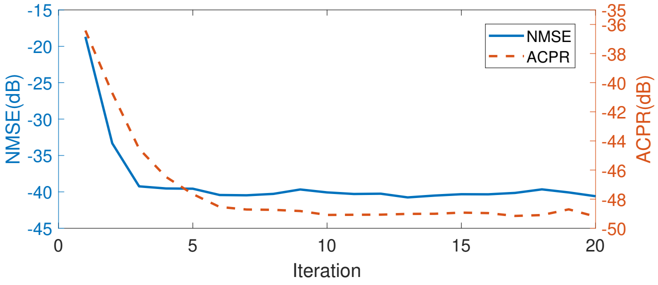

Fig. 2. NMSE and ACPR vs. iteration proving the PCA robustness in DPD linearization.

$$\rho_{CCA} = \frac{\langle \boldsymbol{U}\boldsymbol{p}_i, \boldsymbol{y} \rangle}{||\boldsymbol{U}\boldsymbol{p}_i||_2 ||\boldsymbol{y}||_2}; \tag{4}$$

while PLS maximizes the covariance factor ρ_{PLS}, defined as

$$\rho_{PLS} = \frac{\langle \boldsymbol{U}\boldsymbol{p}_i, \boldsymbol{y} \rangle}{||\boldsymbol{p}_i||_2 ||\boldsymbol{y}||_2} \tag{5}$$

with $\langle \cdot, \cdot \rangle$ being the inner product and $||\cdot||_2$ being the Euclidean norm. Particularizing to our DPD problem, we consider that one set of variables is composed by the DPD basis functions in matrix $\boldsymbol{U}$, and the other one by the signal to be estimated (in case of PA modeling) or to be linearized (in case of DPD linearization), i.e., the vector signal $\boldsymbol{y}$. In essence, CCA finds the directions (or components) of maximum correlation while PLS finds the directions of maximum covariance. The objective is to find the $M \times 1$ vector of coefficients $\boldsymbol{p}_i$, necessary for creating a new component $\boldsymbol{U}\boldsymbol{p}_i$ maximally related to $\boldsymbol{y}$ in terms of maximal correlation (in the case of CCA) or maximal covariance (in the case of PLS). Therefore, in the case of CCA the target can be mathematically defined as

$$\max_{\boldsymbol{p}_i} \{\rho_{CCA}\} = \max_{\boldsymbol{p}_i} \left\{ \frac{\langle \boldsymbol{U}\boldsymbol{p}_i, \boldsymbol{y} \rangle}{||\boldsymbol{U}\boldsymbol{p}_i||_2 ||\boldsymbol{y}||_2} \right\} \tag{6}$$

whereas, in the case of PLS, the target can be defined as follows,

$$\max_{\boldsymbol{p}_i} \{\rho_{PLS}\} = \max_{\boldsymbol{p}_i} \left\{ \frac{\langle \boldsymbol{U}\boldsymbol{p}_i, \boldsymbol{y} \rangle}{||\boldsymbol{p}_i||_2 ||\boldsymbol{y}||_2} \right\} \tag{7}$$

Note that if the matrix $\boldsymbol{U}$ is unitary (i.e., $\boldsymbol{U}^H\boldsymbol{U} = \boldsymbol{I}$), then CCA becomes PLS

$$\begin{aligned} \rho_{CCA} &= \frac{\langle \boldsymbol{U}\boldsymbol{p}_i, \boldsymbol{y} \rangle}{||\boldsymbol{U}\boldsymbol{p}_i||_2 ||\boldsymbol{y}||_2} = \frac{\langle \boldsymbol{U}\boldsymbol{p}_i, \boldsymbol{y} \rangle}{\sqrt{\left(\boldsymbol{p}_i^H\boldsymbol{U}^H\boldsymbol{U}\boldsymbol{p}_i\right)} ||\boldsymbol{y}||_2} \\ &= \frac{\langle \boldsymbol{U}\boldsymbol{p}_i, \boldsymbol{y} \rangle}{||\boldsymbol{p}_i||_2 ||\boldsymbol{y}||_2} = \rho_{PLS} \end{aligned} \tag{8}$$

For those familiar with the conjugate gradient (CG) method, it is worth to mention that the CG is similar to the PLS algorithm. However, while the purpose of using PLS is to create a new transformed basis that presents maximum covariance with the signal to be estimated, the purpose of CG is to perform an iterative search of a set of coefficients that converges to the solution that minimizes a specific quadratic function. As detailed in [27], despite the fact that PLS and CG may have different original goals, both solutions are obtained by equivalent algorithmic procedures.

In the following subsection we will use the precalculated PCA transformation matrix to convert the original data matrix containing the DPD basis functions into a unitary transformed matrix. This way we will be able to maximize the correlation factor as in the CCA but using PLS.

III. DPD Coefficients Adaptation Based on PCA and Dynamic PLS

A. The PCA-DPLS Approach for DPD Linearization

Following a closed-loop direct learning approach [28] as shown in Fig. 1, the DPD coefficients are iteratively updated as follows

$$\boldsymbol{w}_{i+1} = \quad \boldsymbol{w}_i + \Delta\boldsymbol{w} \tag{9}$$

with $\boldsymbol{w}_i$ being the $M \times 1$ vector of coefficients of the DPD model at the i^{th} iteration. The LS solution for $\Delta\boldsymbol{w}$ is

$$\Delta\boldsymbol{w} = \quad \mu(\boldsymbol{U}^H\boldsymbol{U})^{-1}\boldsymbol{U}^H\boldsymbol{e} \tag{10}$$

where μ is a learning-rate parameter. The linearization error vector $\boldsymbol{e}$ is defined as $\boldsymbol{e} = \frac{\boldsymbol{y}}{G_0} - \boldsymbol{u}$, where G_0 is the desired PA linear gain, the $N\times 1$ vectors $\boldsymbol{y}$ and $\boldsymbol{u}$ are the PA output and input signals, respectively; and $\boldsymbol{U}$, defined in (2), is the $N\times M$ data matrix that contains the M basis functions describing the DPD behavioral model.

In the first approach, in order to reduce the number of coefficients in the DPD forward path, the OMP greedy algorithm is applied to select the most relevant basis functions of $\boldsymbol{U}$. Then, in order reduce the number of required DPD parameters in the adaptation path while keeping the same linearization performance, we propose our PCA-DPLS technique as a lower complexity alternative to QR decomposition combined with LS (QR-LS).

The proposed PCA-DPLS technique is described in the following. First, in an off-line process, we calculate the $M\times M$ PCA transformation matrix $\boldsymbol{R}_{PCA}$, that transforms the original DPD basis $\boldsymbol{U}$ into an orthogonal subspace $\tilde{\boldsymbol{U}}$, as described in (3). Then, each column of the transformed basis is normalized as follows,

$$\widehat{\boldsymbol{U}} = \tilde{\boldsymbol{U}}\boldsymbol{T}_{norm} = \boldsymbol{U}\boldsymbol{R}_{PCA}\boldsymbol{T}_{norm} = \boldsymbol{U}\boldsymbol{R} \tag{11}$$

with $\boldsymbol{T}_{norm}$ being a $M\times M$ diagonal matrix composed by the norm of each of the columns of $\tilde{\boldsymbol{U}}$. It is worth to mention that, like with $\boldsymbol{R}_{PCA}$, $\boldsymbol{T}_{norm}$ is also calculated off-line and only once. Therefore, after PCA transformation and normalization, the resulting transformed matrix $\widehat{\boldsymbol{U}}$ is unitary.

After this off-line process where we pre-calculate the transformation matrix $\boldsymbol{R}$, a dynamic PLS approach, which is an upgraded version of the DOTM algorithm presented in [24], is employed to allow dynamic basis selection for the DPD adaptation. The proposed PCA-DPLS approach is described in Algorithm 1. Notice that at every iteration of the adaptation, $\boldsymbol{R}$ is fixed, while $\boldsymbol{U}$ and $\boldsymbol{e}$ are updated with the new input data.

Algorithm 1 PCA-DPLS Calculation

```
procedure PCA-DPLS($\boldsymbol{U}, \boldsymbol{R}, \boldsymbol{e}, \mu, \delta_1, \delta_2$)
    initialization:
    $\boldsymbol{d} = \mu\boldsymbol{e}$;   $\boldsymbol{V}^{(0)} = \{\}$;   $\boldsymbol{P}^{(0)} = \{\}$;   $\boldsymbol{w_{pls}}^{(0)} = \{\}$;
    $j = 0$;   $E_{th1} = \delta_1||\boldsymbol{d}||_2^2$;   $E_{th2} = \delta_2||\boldsymbol{d}||_2^2$;
    $\boldsymbol{r}_1 = (\boldsymbol{UR})^H\boldsymbol{d} = \boldsymbol{R}^H(\boldsymbol{U}^H\boldsymbol{d})$;
    repeat
        $j = j + 1$;
        $\boldsymbol{p}_j = \frac{\boldsymbol{r}_j}{||\boldsymbol{U}(\boldsymbol{R}\boldsymbol{r}_j)||_2}$;
        $\boldsymbol{P}^{(j)} \leftarrow \boldsymbol{P}^{(j-1)} \bigcup \boldsymbol{p}_j$;
        $\boldsymbol{v}_j = \boldsymbol{R}^H\boldsymbol{U}^H\boldsymbol{U}\boldsymbol{R}\boldsymbol{p}_j$;
        for $repeat = 1$ to $2$ do
            for $i = 1$ to $j - 1$ do
                $\boldsymbol{v}_i = \boldsymbol{V}(:, i)$;
                $\boldsymbol{v}_j = \boldsymbol{v}_j - (\boldsymbol{v}_i^H\boldsymbol{v}_j)\boldsymbol{v}_i$;
            end for
        end for
        $\boldsymbol{v}_j = \frac{\boldsymbol{v}_j}{||\boldsymbol{v}_j||_2}$;
        $\boldsymbol{r}_j = \boldsymbol{r}_j - \boldsymbol{v}_j(\boldsymbol{v}_j^H\boldsymbol{r}_j)$;
        $\boldsymbol{V}^{(j)} \leftarrow \boldsymbol{V}^{(j-1)} \bigcup \boldsymbol{v}_j$;
        $\boldsymbol{r}_{j+1} = \boldsymbol{r}_j - \boldsymbol{V}^{(j)}((\boldsymbol{V}^{(j)})^H\boldsymbol{r}_j)$;
        $\widehat{\boldsymbol{x}}_j = \boldsymbol{U}(\boldsymbol{R}\boldsymbol{p}_j)$;
        $w_{pls_j} = \widehat{\boldsymbol{x}}_j^H\boldsymbol{d}$;
        $\boldsymbol{w_{pls}}^{(j)} \leftarrow \boldsymbol{w_{pls}}^{(j-1)} \bigcup w_{pls_j}$;
        $E_j = ||\boldsymbol{w_{pls}}^{(j)}||_2^2$;
    until $(|w_{pls_1}|^2 < E_{th1})$ OR $(E_j > E_{th2})$ OR $(j == dimension(\boldsymbol{U}))$
    $\Delta\widehat{\widehat{\boldsymbol{w}}} = \boldsymbol{w_{pls}}$;
    Return $\boldsymbol{P}, \Delta\widehat{\widehat{\boldsymbol{w}}}$
end procedure
```

With the PCA-DPLS Algorithm, at every iteration of DPD update process we obtain a new PLS transformed matrix $\boldsymbol{P}$ with the minimum necessary number of new components. The criteria to decide the minimum required number of components will be described in the following and is based on the thresholds E_{th1} and E_{th2}, defined as a percentage δ_1 and δ_2 of the energy of the error signal $\boldsymbol{e}$. The transformed basis matrix $\widehat{\boldsymbol{U}}$ obtained from (11) is one more time transformed through the $M \times L$ (where L is variable and may change at every update iteration) transformation matrix $\boldsymbol{P}$ as follows

$$\widehat{\widehat{\boldsymbol{U}}} = \widehat{\boldsymbol{U}}\boldsymbol{P} = \boldsymbol{URP} \tag{12}$$

The new transformed matrix $\widehat{\widehat{\boldsymbol{U}}}$ presents orthonormal components and is $N \times L$ dimensional, where the number of components L is variable as it depends on the dimensions of the transformation matrix $\boldsymbol{P}$.

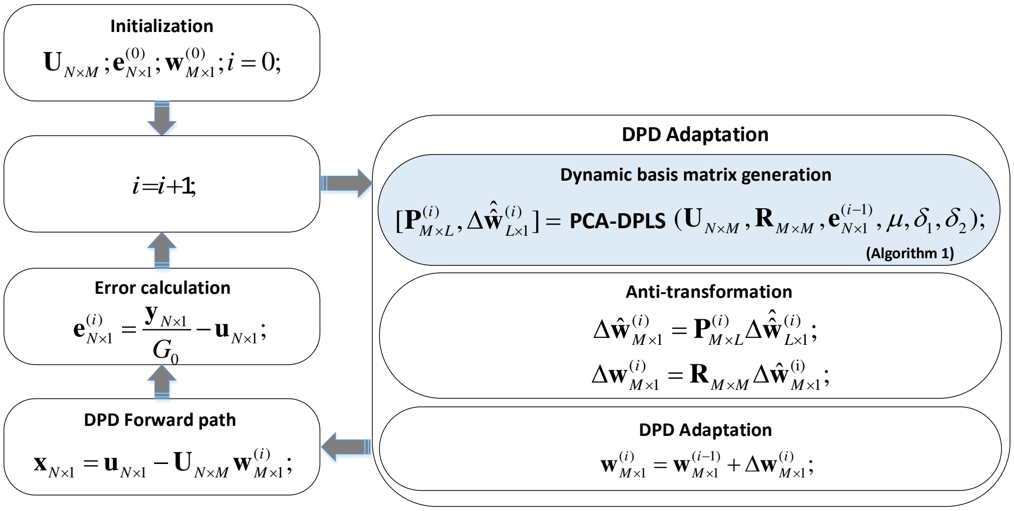
Initialization
$\mathbf{U}_{N\times M};\mathbf{e}^{(0)}_{N\times 1};\mathbf{w}^{(0)}_{M\times 1};i=0;$
$i=i+1;$
Error calculation
$\mathbf{e}^{(i)}_{N\times 1}=\frac{\mathbf{y}_{N\times 1}}{G_0}-\mathbf{u}_{N\times 1};$
DPD Forward path
$\mathbf{x}_{N\times 1}=\mathbf{u}_{N\times 1}-\mathbf{U}_{N\times M}\mathbf{w}^{(i)}_{M\times 1};$
DPD Adaptation
Dynamic basis matrix generation
$[\mathbf{P}^{(i)}_{M\times L},\Delta\hat{\hat{\mathbf{w}}}^{(i)}_{L\times 1}]=$ PCA-DPLS $(\mathbf{U}_{N\times M},\mathbf{R}_{M\times M},\mathbf{e}^{(i-1)}_{N\times 1},\mu,\delta_1,\delta_2);$
(Algorithm 1)
Anti-transformation
$\Delta\hat{\mathbf{w}}^{(i)}_{M\times 1}=\mathbf{P}^{(i)}_{M\times L}\Delta\hat{\hat{\mathbf{w}}}^{(i)}_{L\times 1};$
$\Delta\mathbf{w}^{(i)}_{M\times 1}=\mathbf{R}_{M\times M}\Delta\hat{\mathbf{w}}^{(i)}_{M\times 1};$
DPD Adaptation
$\mathbf{w}^{(i)}_{M\times 1}=\mathbf{w}^{(i-1)}_{M\times 1}+\Delta\mathbf{w}^{(i)}_{M\times 1};$

Fig. 3. Flowchart of the DPD estimation/adaptation using the PCA-DPLS technique.

If the PCA transformation matrix $\boldsymbol{R}_{PCA}$ is perfectly orthogonal and the normalization matrix $\boldsymbol{T}_{norm}$ is perfectly diagonal, then the resulting transformed data matrix $\widehat{\boldsymbol{U}}$ is unitary. In this case, the combination of PCA and dynamic PLS is equivalent to CCA (as justified in Subsection II-C). Therefore, after the PCA-DPLS transformation, the new orthonormal components (i.e., columns) of the matrix $\widehat{\widehat{\boldsymbol{U}}}$ appear sorted according to their contribution to maximize the correlation factor between the new components and the error signal $\boldsymbol{e}$. Having a sorted set of orthonormal components ensures that by selecting the first components we get the most relevant ones in terms of correlation.

However, if the resulting transformed data matrix $\widehat{\boldsymbol{U}}$ is not perfectly unitary, then some degradation in the sorting of the new orthonormal components will exist. In the estimation procedure, this degradation effect is detected and solved by introducing two thresholds in the coefficients extraction process. These two thresholds are set for the purpose of detecting any degradation and decide how many relevant or good coefficients should be calculated, making the DPD estimation more robust and less likely to drift.

- The first threshold, E_{th1}, is determined as a percentage δ_1 of the energy (sum of squares) of the error to be estimated $\boldsymbol{e}$. The energy of the first coefficient is calculated and compared to the threshold E_{th1}. If the first coefficient is not good enough to estimate the error, the threshold is not achieved. In this case, the updating process is stopped and no more coefficients are calculated. This can be decided relying on the fact that the coefficients are sorted. If the first coefficient is good enough to estimate the error, then the threshold is met and more coefficients are calculated until the second threshold is reached.
- The second threshold E_{th2} is determined as a percentage δ_2 of the energy of the error $\boldsymbol{e}$. The energy of all the calculated coefficients (sum of squares) is evaluated and compared to the threshold E_{th2}. The PCA-DPLS algorithm will continue estimating coefficients until the threshold E_{th2} is met.

The key advantage of the PCA-DPLS approach proposed in this paper with respect to the DPLS approach (i.e., the DOTM algorithm) presented in [24] is that the orthonormal components of the transformed matrix $\widehat{\widehat{\boldsymbol{U}}}$ are sorted. Taking advantage of this property, we can select the least number of required coefficients to achieve the given thresholds (E_{th1} and E_{th2}). Whereas in the DPLS approach in [24], since the new components are not properly sorted, it is necessary to estimate more coefficients to reach the same threshold E_{th2}.

Taking into account the orthonormal property of the transformed matrix $\widehat{\widehat{\boldsymbol{U}}}$ (i.e., $\widehat{\widehat{\boldsymbol{U}}}^H\widehat{\widehat{\boldsymbol{U}}}=\boldsymbol{I}$), the update of the transformed DPD coefficients is simplified with respect to (10) as

$$\Delta\widehat{\widehat{\boldsymbol{w}}}=\mu\widehat{\widehat{\boldsymbol{U}}}^H\boldsymbol{e} \tag{13}$$

Finally, the original $\Delta\boldsymbol{w}$ can be found by applying the corresponding anti-transformations

$$\Delta\boldsymbol{w}=\boldsymbol{R}\Delta\widehat{\boldsymbol{w}}=\boldsymbol{R}\boldsymbol{P}\Delta\widehat{\widehat{\boldsymbol{w}}} \tag{14}$$

To sum up, Fig. 3 schematically describes the proposed PCA-DPLS approach for DPD estimation/adaptation.

B. Computational Complexity

In order to estimate the computational cost of the proposed PCA-DPLS adaptation algorithm, we compare it to both the QR decomposition using Givens rotation and the CCA. Both approaches have a computational complexity of $\mathcal{O}(NM^2)$ as explained in [29], [30]. The order of magnitude of the computational cost of the PLS reported in [29], [31] is $\mathcal{O}(NML)$, where N is the number of samples, M is the number of basis functions (and thus coefficients) in the forward path and L is the number of PLS components that have been dynamically selected (at every iteration of the DPD update) to carry out the DPD coefficient's update.

Despite the fact that the PCA transformation matrix is calculated off-line only once, the computational complexity of the PCA algorithm is analyzed in the following. The

PCA algorithm comprises two main steps: (i) calculating the covariance matrix, with computational cost $\mathcal{O}(NM^2)$; and (ii) generating the eigenvalue decomposition of the covariance matrix, with computational cost $\mathcal{O}(M^3)$. Thus, the total computational cost of PCA is $O(NM^2 + M^3)$ [32].

In addition, the proposed PCA-DPLS adaptation approach requires applying two anti-transformations (as defined in (14) or shown in the flowchart in Fig. 3), where the computational cost of each of them is $\mathcal{O}(ML)$ and $\mathcal{O}(M^2)$, respectively.

Hence, assuming that there is no computational complexity associated to the calculation of the PCA transformation matrix (since it is pre-calculated off-line only once), the computational complexity of the proposed PCA-DPLS algorithm is approximately $\mathcal{O}(NML + M^2 + ML)$. Consequently, if $L \ll M$, the proposed PCA-DPLS approach will introduce significantly less computational complexity than the QR-LS and the CCA techniques. This computational complexity trend will be further analyzed in section IV. A comparison of our PCA-DPLS adaptation algorithm with Matlab's backslash (see Fig. 7) in terms of tic-toc computational time will support the idea that when $L \ll M$, the PCA-DPLS can be more computationally efficient than the QR-LS, since QR-LS always computes M coefficients.

IV. Experimental Testbench and Results

The dynamic DPD parameter update scheme using the PCA-DPLS method has been experimentally evaluated considering the complex waveform scenario and the test setup detailed in [20]. Four 20-MHz bandwidth and 64-QAM modulated fast-convolution filter bank multi-carrier (FC-FBCM) signals, each with different sub-carrier group deactivation configurations, have been carrier aggregated. The overall test signals feature 80 MHz bandwidth and around 13 dB PAPR. Considering a DPD expansion factor by three, the DPD baseband waveform length is of 737280 samples.

The Matlab-controlled digital linearization test bench is shown in Fig. 4. For signal generation and data capture we used commercial boards from Texas Instruments (TI) (i.e., TI TSW1400EVM and TSW30H84EVM at Tx side and TI ADC32RF45EVM and TSW14J56EVM at Rx side). The device under test was a class-J PA based on the Cree CGH35030F GaN HEMT operated at the RF frequency of 875 MHz and delivering 28 dBm of mean output power.

In order to prove the advantages of the proposed PCA-DPLS approach for DPD coefficient estimation/adaptation, we have compared it with the following techniques: the dynamic PLS (DOTM algorithm) introduced in [24], the CCA and the QR-LS. The comparison is established in terms of linearization performance by evaluating the NMSE and the ACPR after DPD linearization, the minimum number of required coefficients in the adaptation DPD subsystem to meet the linearity specifications and the computational running time according to Matlab's tic-toc measurements.

The original basis functions were generated by using the GMP behavioral model with 322 coefficients. By applying the OMP feature selection algorithm, we cut the number of required coefficients in the forward path down to 100. The validity of the proposed PCA-DPLS method is universal, i.e., it does not depend on a specific PA. However, if we had used a different PA, the basis functions in matrix $\boldsymbol{U}$ describing the PA behavior under certain operating conditions (signal bandwidth, level of PA saturation) would have changed.

TABLE I
DPD Performance Comparison.

Configuration 80 MHz FC-FBMC	No. of Coeff. (max/min)	NMSE [dB]	ACPR [dBc]	EVM [%]
No DPD	-	-18.6	-36.35	5.73
QR-LS	100/100	-40.39	-49.08	1.17
DPLS (DOTM)	100/1	-39.77	-49.27	1.19
CCA	1/1	-40.73	-49.34	1.16
PCA-DPLS	10/1	-40.35	-49.33	1.17

Fig. 5 and Fig. 6 show the NMSE and ACPR evolution when considering different adaptation methods and taking into account different number of components (and thus coefficients) for the DPD estimation/update. All the DPD techniques under comparison (i.e., the proposed PCA-DPLS, the DPLS based on the DOTM algorithm, the CCA and the QR-LS) converge to around -40 dB of NMSE after 5 iterations and -48 dBc of ACPR after approximately 10 iterations (see Table I). However, as shown in Fig. 5 and Fig. 6, whereas the QR-LS needs a fixed amount of coefficients (up to 100 coefficients) for each iteration of the DPD update, both DPLS (DOTM algorithm) and PCA-DPLS dynamically select the minimum necessary components to reach the targeted linearity levels. Finally, with the CCA technique, only 1 coefficient at every DPD adaptation iteration was necessary to meet the linearity specifications. As shown in Table I, the main difference between PCA-DPLS and DPLS is that, by including the PCA transformation, the dynamic selection of coefficients with PCA-DPLS is more efficient (i.e., the same performance is reached with less coefficients). Therefore, unlike in DPLS where the algorithm dynamically selects among, for example, 100, 54, 27, 7 and 1 coefficient(s) depending on the iteration as shown in Fig. 5 and Fig. 6, with PCA-DPLS, most of the times only 1 coefficient is necessary and only when the transformed matrix $\widehat{\boldsymbol{U}}$ is not perfectly unitary, the algorithm selects more coefficients (e.g., up to 10). Fig. 7 shows both the unlinearized spectrum and the linearized spectrum when considering 100 coefficients in the DPD forward path and 1 coefficient in the feedback path with PCA-DPLS adaptation.

As explained in Subsection III-A, after the PCA-DPLS transformation, the new orthonormal components of the matrix $\widehat{\widehat{\boldsymbol{U}}}$ (and also the coefficients) appear sorted according to their contribution to maximize the correlation factor between the new components and the error signal $\boldsymbol{e}$. This is the key advantage of the proposed PCA-DPLS with respect to DPLS. Fig. 8 depicts the magnitude of the DPD coefficients when applying DPLS (Fig. 8-top) and PCA-DPLS (Fig. 8-bottom). Although in both cases the general trend shows that the magnitude of the coefficients decreases, in the case of PCA-DPLS the sorting is more accurate, i.e., close to monotonically decreasing, which allows avoiding the estimation of several

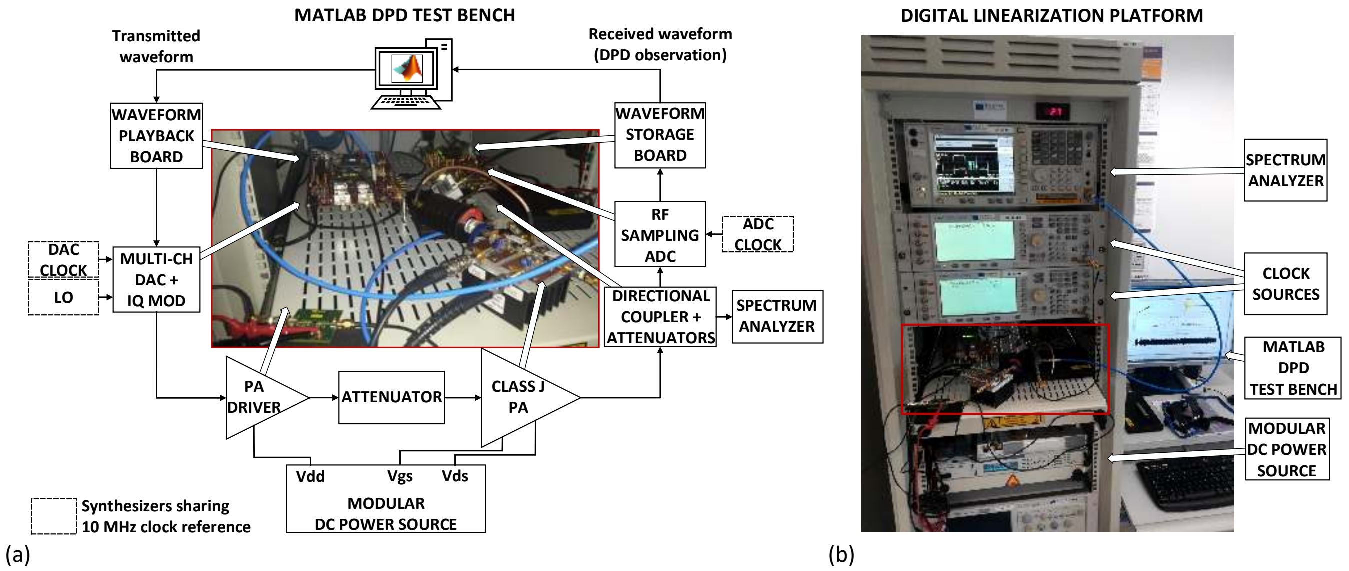

Fig. 4. (a) Block diagram of the DPD linearization test bench used for experimental validation including a picture of the PA used; (b) Picture of the overall Matlab-controlled digital linearization platform including the laboratory instrumentation used. ADC: analog-to-digital converter, CH: channel, DAC: digital-to-analog converter, Vdd: drain-to-drain voltage, Vds: drain-to-source voltage, Vgs: gate-to-source voltage, IQ MOD: in-phase and quadrature-phase modulator, LO: local oscillator. Adapted from [20].

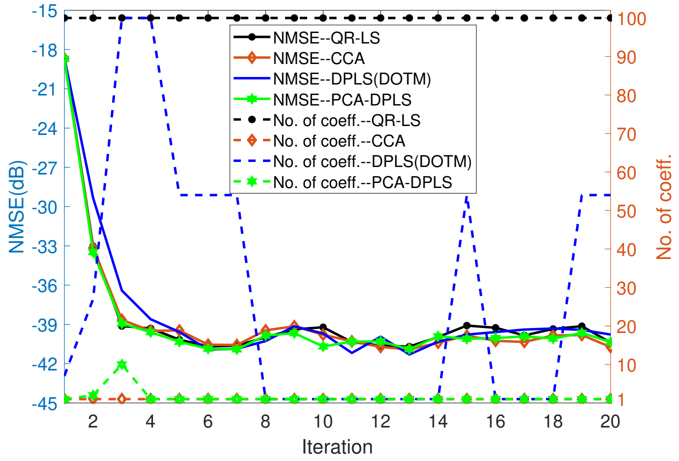

Fig. 5. Evolution of the NMSE for PCA-DPLS, DPLS, CCA and QR-LS.

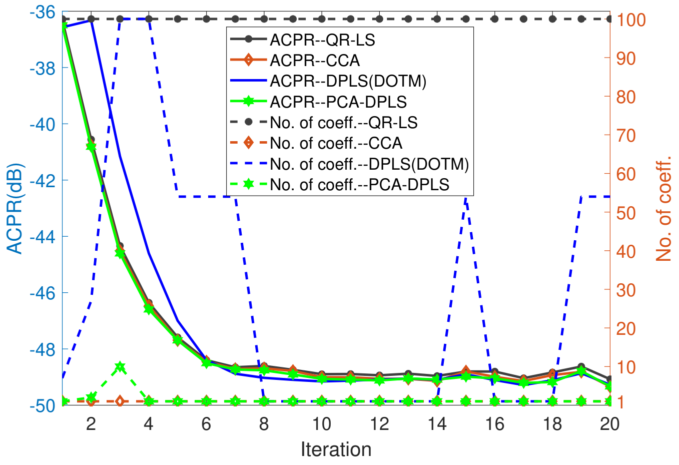

Fig. 6. Evolution of the ACPR for PCA-DPLS, DPLS, CCA and QR-LS.

unnecessary coefficients in comparison to DPLS.

The advantage of the PCA-DPLS technique is also validated in terms of the Matlab's tic-toc processing time. Taking as a reference the processing time (tic-toc) of Matlab's backslash operation ("\") with 100 coefficients, Fig. 9 shows the relative factors of the processing time when considering PCA-DPLS using 1, 10 and 100 coefficients. It can be seen that, when considering the same number of estimated coefficients (i.e., 100 coefficients), the Matlab's backslash operation is around 2 times faster than the proposed PCA-DPLS. However, the proposed PCA-DPLS will significantly reduce the number of computed coefficients in the DPD adaptation subsystem while still achieving the same linearity levels than QR-LS. Therefore, by significantly reducing the number of coefficients, for example, down to 10 coefficients, the PCA-DPLS processing time is only one third that of Matlab's backslash operation. Moreover,

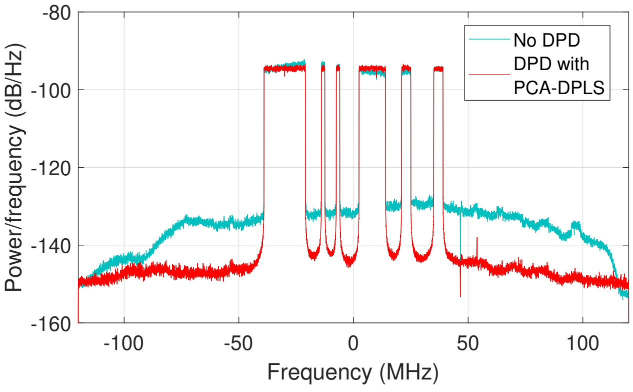

Fig. 7. Spectra of the PA output before and after DPD linearization using the PCA-DPLS technique.

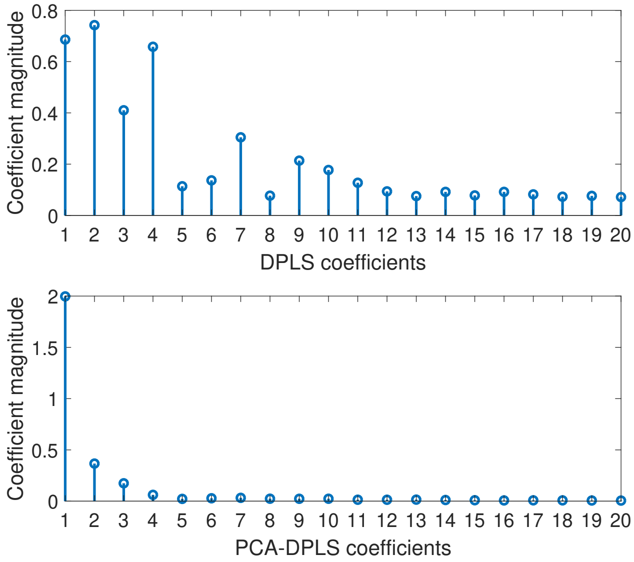

Fig. 8. Magnitude of the DPLS (DOTM) coefficients (top) and magnitude of the PCA-DPLS coefficients (bottom) at DPD iteration update 3.

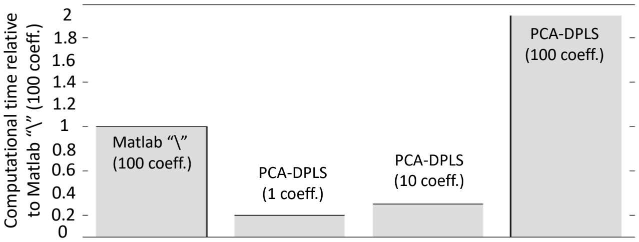

Fig. 9. Computational time of the PCA-DPLS algorithm considering different number of coefficients and taking as a reference the computational time of Matlab's backslash operation.

in the case of using only 1 coefficient (which happens with high probability since PCA-DPLS is equivalent to CCA when no significant degradation occurs), the PCA-DPLS running time is five times faster than Matlab's backslash operation.

V. Conclusion

In this paper we proposed a new technique for dynamically estimating and updating the DPD coefficients based on the combination of the PCA transformation with the PLS extraction of components. The proposed PCA-DPLS approach significantly improves the model order reduction capabilities of the DPLS technique (DOTM) proposed by the authors in [24] and is equivalent to a CCA updating solution, which is optimal in the sense of generating components with maximum correlation. The PCA-DPLS method allows to update as many coefficients as necessary for achieving the required linearity, and to stop this update when it detects that the DPD basis is not able to estimate and minimize the remaining nonlinear error. This allows to reduce the computational cost and to overcome ill-conditioning problems in comparison to other methods that use a fixed number of coefficients when solving the required LS estimation in the DPD adaptation loop. The proposed dynamic adaptation technique has been tested and compared in terms of linearization performance and computational cost with the commonly used QR decomposition approach for solving the LS problem. Experimental results show that the proposed PCA-DPLS method drastically reduces the amount of DPD coefficients required in the DPD adaptation subsystem while maintaining the same linearization performance, which ultimately impacts the computational cost and running time.

Acknowledgment

The authors would like to thank Dr. J. A. García, Dr. M. N. Ruiz, Dr. J. R. Pérez-Cisneros and Mr. D. Vegas from Universidad de Cantabria for the donation of the class-J power amplifier used in this paper.

References

[1] Z. Wang, "Demystifying envelope tracking: Use for high-efficiency power amplifiers for 4g and beyond," *IEEE Microwave Magazine*, vol. 16, no. 3, pp. 106–129, apr 2015.

[2] R. Pengelly, C. Fager, and M. Ozen, "Doherty's legacy: A history of the doherty power amplifier from 1936 to the present day," *IEEE Microwave Magazine*, vol. 17, no. 2, pp. 41–58, feb 2016.

[3] R. Quaglia and S. Cripps, "A load modulated balanced amplifier for telecom applications," *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, no. 3, pp. 1328–1338, March 2018.

[4] Z. Popovic and J. A. Garcia, "Microwave class-e power amplifiers: A brief review of essential concepts in high-frequency class-e PAs and related circuits," *IEEE Microwave Magazine*, vol. 19, no. 5, pp. 54–66, jul 2018.

[5] P. L. Gilabert and G. Montoro, "3-D Distributed Memory Polynomial Behavioral Model for Concurrent Dual-Band ET PA Linearization," *IEEE Trans. on Microw. Theory and Tech.*, vol. 63, pp. 638–648, Feb 2015.

[6] 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.

[7] P. L. Gilabert, G. Montoro, et al., "Comparison of model order reduction techniques for digital predistortion of power amplifiers," in *46th European Microw. Conf. (EuMC)*, Oct. 2016, pp. 182–185.

[8] P. L. Gilabert, D. Lopez-Bueno, and G. Montoro, "Spectral weighting orthogonal matching pursuit algorithm for enhanced out-of-band digital predistortion linearization," *IEEE Transactions on Circuits and Systems II: Express Briefs*, pp. 1–1, 2018.

[9] 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.

[10] P. L. Gilabert, A. Cesari, G. Montoro, E. Bertran, and J. M. Dilhac, "Multi Look-Up Table FPGA Implementation of a Digital Adaptive Predistorter for Linearizing RF Power Amplifiers with Memory Effects," *IEEE Trans. Microw. Theory Techn.*, vol. 56, no. 2, pp. 372–384, Feb. 2008.

[11] A. Molina, K. Rajamani, and K. Azadet, "Concurrent dual-band digital predistortion using 2-D lookup tables with bilinear interpolation and extrapolation: Direct least squares coefficient adaptation," *IEEE Transactions on Microwave Theory and Techniques*, vol. 65, no. 4, pp. 1381–1393, Apr. 2017.

[12] N. Mrabet, I. Mohammad, F. Mkadem, C. Rebai, and S. Boumaiza, "Optimized hardware for polynomial digital predistortion system implementation," in *2012 IEEE Topical Conf. on Power Amplifiers for Wireless and Radio Appl. (PAWR)*. IEEE, Jan. 2012, pp. 83–84.

[13] W. Cao, Y. Li, and A. Zhu, "Magnitude-selective affine function based digital predistorter for RF power amplifiers in 5g small-cell transmitters," in *2017 IEEE MTT-S International Microwave Symposium (IMS)*. IEEE, jun 2017.

[14] D. Wisell, J. Jalden, and P. Handel, "Behavioral power amplifier modeling using the lasso," in *2008 IEEE Inst. and Meas. Tech. Conf.*, May 2008, pp. 1864–1867.

[15] L. Guan and A. Zhu, "Optimized low-complexity implementation of least squares based model extraction for digital predistortion of RF power amplifiers," *IEEE Trans. Microw. Theory Techn.*, vol. 60, no. 3, pp. 594–603, Mar. 2012.

[16] J. Peng, S. He, B. Wang, Z. Dai, and J. Pang, “Digital predistortion for power amplifier based on sparse bayesian learning,” *IEEE Trans. on Circ. and Sys. II: Express Briefs*, vol. 63, no. 9, pp. 828–832, Sep. 2016.

[17] 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.

[18] P. Geladi and B. R. Kowalski, “Partial least-squares regression: a tutorial,” *Analytica Chimica Acta*, vol. 185, pp. 1–17, 1986.

[19] H. Hotelling, “Relations Betwen Two Sets of Variates,” *Biometrika*, vol. 28, no. 3-4, pp. 321–377, dec 1936.

[20] D. Lopez-Bueno, Q. A. Pham, G. Montoro, and P. L. Gilabert, “Independent digital predistortion parameters estimation using adaptive principal component analysis,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, no. 12, pp. 5771–5779, dec 2018.

[21] Q. A. Pham, D. López-Bueno, G. Montoro, and P. L. Gilabert, “Adaptive principal component analysis for online reduced order parameter extraction in PA behavioral modeling and DPD linearization,” in *2018 IEEE MTT-S Int. Microw. Symp. (IMS)*, Jun. 2018, pp. 160–163.

[22] Q. A. Pham, D. Lopez-Bueno, T. Wang, G. Montoro, and P. L. Gilabert, “Partial least squares identification of multi look-up table digital predistorters for concurrent dual-band envelope tracking power amplifiers,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, no. 12, pp. 5143–5150, dec 2018.

[23] S. Muruganathan and A. Sesay, “A QRD-RLS-based predistortion scheme for high-power amplifier linearization,” *IEEE Trans. Circuits Syst. II, Express Briefs*, vol. 53, no. 10, pp. 1108–1112, Oct. 2006.

[24] Q. A. Pham, D. López-Bueno, G. Montoro, and P. L. Gilabert, “Dynamic selection and update of digital predistorter coefficients for power amplifier linearization,” in *Proc. 2019 IEEE Topical Conf. on RF/Microw. Power Amplifiers for Radio and Wireless Appl. (PAWR)*, Jan. 2019, pp. 1–4.

[25] 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.

[26] R. Rosipal and N. Krämer, “Overview and recent advances in partial least squares,” in *Subspace, Latent Structure and Feature Selection*. Springer Berlin Heidelberg, 2006, pp. 34–51.

[27] A. Phatak and F. de Hoog, “Exploiting the connection between PLS, lanczos methods and conjugate gradients: alternative proofs of some properties of PLS,” *Journal of Chemometrics*, vol. 16, no. 7, pp. 361–367, 2002.

[28] 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.

[29] 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.

[30] J. Chen, G. Wang, Y. Shen, and G. B. Giannakis, “Canonical correlation analysis of datasets with a common source graph,” *IEEE Transactions on Signal Processing*, vol. 66, no. 16, pp. 4398–4408, aug 2018.

[31] B. V. Srinivasan, W. R. Schwartz, R. Duraiswami, and L. Davis, “Partial least squares on graphical processor for efficient pattern recognition,” Tech. Rep., 2010.

[32] A. Sharma and K. K. Paliwal, “Fast principal component analysis using fixed-point algorithm,” *Pattern Recognition Letters*, vol. 28, no. 10, pp. 1151–1155, jul 2007.

Quynh Anh Pham (S’18) received the M.Sc. degree in Computer Science and Engineering from the Soongsil University, Seoul, South Korea, in 2015. Since 2017, she has been with the department of Signal Theory and Communications (TSC) at Universitat Politècnica de Catalunya (UPC), Barcelona, Spain where she is currently pursuing the Ph.D. degree. Her research interests include wireless communications, nonlinear behavioral modeling and linearization techniques for high-efficiency transmitters in wireless communication systems.

Gabriel Montoro (M’09) received the M.Sc. degree in telecommunication engineering and the Ph.D. degree from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 1990 and 1996, respectively. He joined the Department of Signal Theory and Communications (TSC) in 1991, where he is currently an Associate Professor. His first research works were done on the area of adaptive control, and now his main research interest is in the use of signal processing strategies for power efficiency improvement in communications systems.

David López–Bueno (M’13) received his M.Sc. degree in Telecommunication Engineering from the Universitat Politècnica de Catalunya (UPC) in 2005. Starting 2003, he worked for four years as RF module & MMIC Design Engineer in space companies. By late 2006, he joined the CTTC as Research Engineer and since then he has participated in a number of research projects dealing with RF transceivers, 4G/5G MIMO OFDM-based lab demonstrators, and assessing the risk of interference between heterogeneous systems. Since 2013, he is also working in projects in the field of wideband PA digital linearization for radio equipment which has led him to pursue a Phd on this topic at UPC–Barcelona Tech.

Pere L. Gilabert (M’09-SM’13) received his M.Sc. degree in Telecommunication Engineering from the Universitat Politècnica de Catalunya (UPC) in 2002, and he developed his Master Thesis at the University of Rome “La Sapienza” with an Erasmus exchange grant. He joined the department of Signal Theory and Communications (TSC) in 2003 and received his Ph.D., awarded with the Extraordinary Doctoral Prize, from the UPC in 2008. He is an Associate Professor at the Castelldefels School of Telecommunications and Aerospace Engineering (EETAC-UPC). His research activity is in the field of linearization techniques and digital signal processing solutions for highly efficient transmitter architectures.