

Independent Digital Predistortion Parameters Estimation Using Adaptive Principal Component Analysis

David López-Bueno^{ID}, *Member, IEEE*, Quynh Anh Pham^{ID}, *Student Member, IEEE*,
Gabriel Montoro^{ID}, *Member, IEEE*, and Pere L. Gilabert^{ID}, *Senior Member, IEEE*

Abstract—This paper presents an estimation/adaptation method based on the adaptive principal component analysis (APCA) technique to guarantee the identification of the minimum necessary parameters of a digital predistorter. The proposed estimation/adaptation technique is suitable for online field-programmable gate array or system on chip implementation. By exploiting the orthogonality of the resulting transformed matrix obtained with the APCA technique, it is possible to reduce the number of coefficients to be estimated which, at the same time, has a beneficial regularization effect by preventing ill-conditioning or overfitting problems. Therefore, this identification/adaptation method enhances the robustness of the parameter estimation and simplifies the adaptation by reducing the number of estimated coefficients. Due to the orthogonality of the new basis, these parameters can be estimated independently, thus allowing for scalability. Experimental results will show that it is possible to determine the minimum number of parameters to be estimated in order to meet the targeted linearity levels while ensuring a robust well-conditioned identification. Moreover, the results will show how thanks to the orthogonality property of the new basis functions, the coefficients of the digital predistorter can be estimated independently. This allows to tradeoff the digital predistorter adaptation time versus performance and hardware complexity.

Index Terms—Digital predistortion (DPD), linearization, machine learning, power amplifier (PA), principal component analysis (PCA).

I. INTRODUCTION

DIGITAL predistortion (DPD) linearization is the most common and spread solution to cope with the power

Manuscript received May 2, 2018; revised July 28, 2018; accepted August 28, 2018. This work was supported in part by the Spanish Government (MINECO) and FEDER under Project TEC2014-58341-C4-03/04-R and Project TEC2017-83343-C4-2-R and in part by the Generalitat de Catalunya under Grant 2017 SGR 891. This paper is an expanded version from the 2018 IEEE MTT-S International Microwave Symposium Conference, Philadelphia, PA, USA, June 10–15, 2018. (*Corresponding author: David López-Bueno.*)

D. López-Bueno is with the Centre Tecnològic de Telecomunicacions de Catalunya, 08860 Barcelona, Spain, and also with the Department of Signal Theory and Communications, Universitat Politècnica de Catalunya, Barcelona Tech, 08034 Barcelona, Spain (e-mail: david.lopez@cttc.es).

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

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TMTT.2018.2870420

amplifier's (PA) inherent linearity versus efficiency tradeoff. The use of fifth-generation new radio (5G-NR) spectrally efficient signals with high peak-to-average power ratios (PAPRs) occupying wider bandwidths only aggravates such compromise. When considering wideband signals, carrier aggregation, or multiband configurations in high-efficient amplification architectures (i.e., Doherty PAs, envelope tracking PAs, or outphasing transmitters) in multiple-input multiple-output transmitters, the number of parameters required by the DPD behavioral model to compensate for the unwanted nonlinearities and memory effects can be unacceptably high.

When targeting a field-programmable gate array (FPGA) implementation, 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. A lot of effort has been dedicated in the literature to propose efficient DPD architectures, either polynomial based [1] or lookup table (LUT) based [2]–[4], in order to implement the forward path of the digital predistorter in a real-time digital signal processor.

The processing capabilities and the memory of FPGAs have increased dramatically over time. Despite this fact, having to deal with parametric models with a huge number of coefficients is not desirable, not only because of its negative impact on the FPGA resources utilization but also because of its extraction/adaptation may derive to overfitting and uncertainty. By applying regularization techniques [5], it is possible to both avoid the numerical ill-conditioning of the estimation and reduce the number of coefficients of the DPD function which ultimately impacts the baseband processing computational complexity and power consumption.

The objective of regularization techniques is to enforce the sparsity constraint on the vector of parameters by minimizing the number of active components (ℓ_0 -norm) subject to a constraint on the two-norm squared of the identification error. Unfortunately, this is a nondeterministic polynomial-time hard (NP-hard) combinatorial search problem. Therefore, in the field of DPD linearization, several suboptimal approaches have been proposed targeting both robust identification and model order reduction, such as: the least absolute shrinkage and selection operator, consisting in a ℓ_1 -norm regularization [6]; the Ridge regression, consisting in a ℓ_2 -norm regularization [7]; the sparse Bayesian learning algorithm [8]; or the

orthogonal matching pursuit (OMP), a greedy algorithm for sparse approximation [4], [9] to select the most relevant basis functions of the DPD function.

However, little attention is paid to the identification/adaptation subsystem that most of the time is addressed in the literature by simply solving the least squares (LS) regression applying the Moore–Penrose inverse to extract the DPD parameters. In addition, in order to deal with an overdetermined system or an ill-conditioned covariance matrix, most of the times the straightforward solution used in the literature consists in using MATLAB’s backslash operator (otherwise known as the “mldivide” function) that employs a QR solver for dense nonsquare matrices like those typically found in DPD. Either in commercial products or in publications addressing FPGA implementation, one of the most common solutions to avoid calculating the inverse of the covariance matrix is extracting the parameters through QR factorization combined with recursive LS [10]. Wang *et al.* [11] present a method to reduce the computational complexity of the direct learning architecture by making use of the constant covariance matrix to avoid recalculating the time-varying matrix for OFDM-like input signals seen as stationary ergodic random processes. The approach and the complexity versus convergence benchmarking are worth considering but, in our work, we take benefit from the independence of the transformed basis in the feedback path to propose an online scalable solution to meet the requirements at the lowest usable complexity.

In [12], we proposed the parameter extraction of the DPD behavioral models using a new method based on the adaptive principal component analysis (APCA) technique. The APCA approach resembles the singular value decomposition but the main difference relies on the fact that APCA is able to continuously track and adapt to the evolution of the eigenvectors required for doing DPD [13]. The proposed block deflated (BD-APCA) technique is a modification of the complex domain generalized Hebbian algorithm [14]. In this paper, we will prove that thanks to the APCA technique, we can iteratively find a new orthogonal basis that will solve the covariance matrix inversion problem by converting the original LS regression into a set of independent adaptive least mean square (LMS) identifications. This process can run online in an FPGA for not only ensuring a proper well-conditioned estimation but also allowing to reduce the number of parameters in the identification process. In [12], we focused on the BD-APCA algorithm description for DPD identification and provided a preliminary experimental validation of the method applicability to DPD by considering an offline precalculated transformation matrix delivering 60 independently calculated coefficients in a row. This paper expands the applicability of the PCA to DPD demonstrated in [15]. Unlike in [15], in this paper, we do not assume that the transformation matrix is calculated only once, offline, and then used for all the data having the same statistical properties as the one used to extract the transformation matrix. Instead, we go a step further toward a possible online implementation of the APCA by demonstrating (both theoretically and experimentally) the benefits of the independent calculation of the DPD coefficients. Alleviating the data processing implementation bottlenecks of the PCA technique, the scalable APCA method is in this paper further

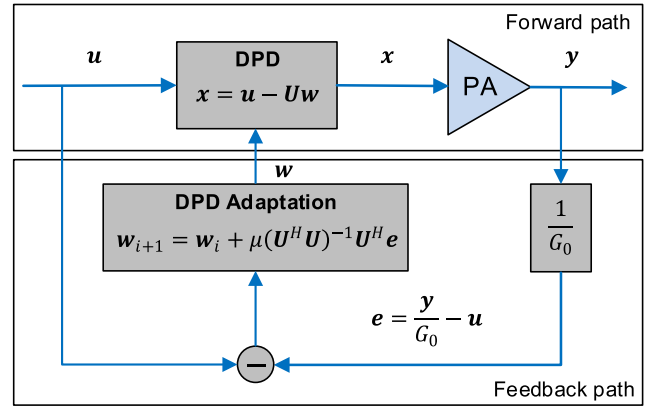


Fig. 1. Block diagram of a closed-loop DPD system following a direct learning approach.

experimentally validated for DPD application by employing an increasing transformation matrix which produces an increasing set of independently calculated coefficients only if the system performance requirements are not yet met (i.e., starting with one coefficient and increasing the number of coefficients by one at every iteration).

Therefore, the remainder of this paper is organized as follows. To create a self-contained paper for the reader, Section II summarizes the basic principles of the closed-loop DPD and the PCA theory. In Section III, the independent identification/adaptation subsystem based on the APCA method, to further reduce the number of the parameters required for a robust identification, is presented. Section IV describes the experimental test bench, while Section V shows experimental results proving the advantages of the proposed independent DPD estimation using the APCA algorithm. Finally, the conclusions are given in Section VI.

II. CLOSED-LOOP DIGITAL PREDISTORTION LINEARIZATION

A. Digital Predistortion Forward Path

The block diagram of our closed-loop adaptive DPD architecture is shown in Fig. 1. In comparison to the indirect learning approach, with the direct learning estimation, we gain robustness against noisy PA output observations and avoid the offset of the coefficient vector from its optimal value [16]. In the forward path, the input–output relationship at the DPD block can be described as

$$x[n] = u[n] - d[n] \quad (1)$$

where $x[n]$ is the signal at the output of the DPD block, $u[n]$ is the input signal, and $d[n]$ is the distortion signal which is described in this paper through the generalized memory polynomial (GMP) behavioral model as defined in [17]

$$d[n] = \sum_{i=0}^{N_a-1} \sum_{p=0}^{P_a-1} \alpha_{pi} \cdot u[n - \tau_i^a] |u[n - \tau_i^a]|^p \times \sum_{j=1}^{M_b} \sum_{i=0}^{N_b-1} \sum_{p=1}^{P_b-1} \beta_{pij} \cdot u[n - \tau_i^b] |u[n - \tau_i^b - \tau_j^b]|^p \times \sum_{j=1}^{M_c} \sum_{i=0}^{N_c-1} \sum_{p=1}^{P_c-1} \gamma_{pij} \cdot u[n - \tau_i^c] |u[n - \tau_i^c + \tau_j^c]|^p. \quad (2)$$

In a more compact notation, (1) can be rewritten as

$$x[n] = u[n] - \boldsymbol{\varphi}^T[n]\mathbf{w}[n] \quad (3)$$

where $\mathbf{w}[n] = (w_1[n], \dots, w_i[n], \dots, w_M[n])^T$ is a vector of coefficients at time n with dimensions $M \times 1$, where M is the order of the behavioral model, and $\boldsymbol{\varphi}^T[n] = (\varphi_1[n], \dots, \varphi_i[n], \dots, \varphi_M[n])$ is the vector containing the basis functions $\varphi_i[n]$, with $i = 1, \dots, M$. By taking into account the GMP behavioral model in (2), the order of the behavioral model is defined as $M = N_a P_a + M_b N_b (P_b - 1) + M_c N_c (P_c - 1)$. The mapping between the GMP specific coefficients $(\alpha_{pi}, \beta_{pij}, \gamma_{pij})$ in (2) and the general purpose DPD coefficients $w_i[n]$ in (3) is straightforward.

Finally, when expressed in matrix notation, (1) can be rewritten as

$$\mathbf{x} = \mathbf{u} - \mathbf{U}\mathbf{w} \quad (4)$$

where $\mathbf{x} = (x[0], \dots, x[n], \dots, x[N-1])^T$ and $\mathbf{u} = (u[0], \dots, u[n], \dots, u[N-1])^T$, with $n = 0, \dots, N-1$, are the predistorted and input vectors, respectively, and $\mathbf{U} = (\boldsymbol{\varphi}[0], \dots, \boldsymbol{\varphi}[n], \dots, \boldsymbol{\varphi}[N-1])^T$ is the $N \times M$ data matrix, with N being the number of samples and M being the number of basis functions or the order of the model.

The DPD function in the forward path can be implemented in a programmable logic (PL) device following either an LUT interpolation approach as in [2] or by using complex multipliers following a polynomial approach using Horner's rule [1], for example. The implementation and order reduction of the DPD function in the forward path has been addressed by the authors in previous publications, and thus falls out of the main scope of this paper.

B. Principal Component Analysis

The PCA theory is used to generate a new basis set of orthogonal components, as explained in [15], which is obtained through a change of basis using a transformation matrix \mathbf{V} that contains the eigenvectors of the covariance matrix of \mathbf{U}

$$\text{cov}(\mathbf{U}) = \frac{1}{N-1} ((\mathbf{U} - E\{\mathbf{U}\})^H (\mathbf{U} - E\{\mathbf{U}\})) \approx \mathbf{U}^H \mathbf{U} \quad (5)$$

where $E\{\cdot\}$ is the expected value. The principal components of the basis functions (i.e., columns of \mathbf{U}) are the eigenvectors of $\mathbf{U}\mathbf{U}^H$. However, as it will be proved, $\mathbf{U}^H \mathbf{U}$ and $\mathbf{U}\mathbf{U}^H$ have the same eigenvalues and, moreover, their eigenvectors are related as described in the following:

$$(\mathbf{U}^H \mathbf{U})\mathbf{v}_i = \lambda_i \mathbf{v}_i \rightarrow (\mathbf{U}\mathbf{U}^H)\mathbf{U}\mathbf{v}_i = \lambda_i \mathbf{U}\mathbf{v}_i \quad (6)$$

with \mathbf{v}_i being the i th eigenvector of $\mathbf{U}^H \mathbf{U}$. For each i

$$(\mathbf{U}^H \mathbf{U})\mathbf{V} = \boldsymbol{\lambda}\mathbf{V} \rightarrow (\mathbf{U}\mathbf{U}^H)\mathbf{U}\mathbf{V} = \boldsymbol{\lambda}\mathbf{U}\mathbf{V} \quad (7)$$

where $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_i, \dots, \mathbf{v}_L)$ is the $M \times L$ transformation matrix, where $L \leq M$. The linear combination $\mathbf{U}\mathbf{V}$ corresponds to the eigenvectors of the matrix $\mathbf{U}\mathbf{U}^H$, which are the desired principal components of the basis functions (i.e., columns) of \mathbf{U} . Moreover, $\boldsymbol{\lambda}$ is the diagonal matrix

containing the eigenvalues of both the $\mathbf{U}\mathbf{U}^H$ and the $\mathbf{U}^H \mathbf{U}$ matrices. Therefore, the new transformed matrix is found as

$$\hat{\mathbf{U}} = \mathbf{U}\mathbf{V} \quad (8)$$

with $\hat{\mathbf{U}} = (\boldsymbol{\psi}[0], \dots, \boldsymbol{\psi}[n], \dots, \boldsymbol{\psi}[N-1])^T$ being the $N \times L$ data matrix, where $\boldsymbol{\psi}^T[n] = (\psi_1[n], \dots, \psi_j[n], \dots, \psi_L[n])$ is the $1 \times L$ data vector containing the new orthogonal basis functions $\psi_j[n]$ (with $j = 1, \dots, L$) at time n .

C. Digital Predistortion Adaptation Path

Following a closed-loop error minimization technique as in [18], the coefficients can be extracted iteratively finding the LS solution. At the n th iteration (i.e., when considering buffers of N data samples), the coefficients are obtained as

$$\mathbf{w}[n+1] = \mathbf{w}[n] + \mu (\mathbf{U}^H \mathbf{U})^{-1} \mathbf{U}^H \mathbf{e} \quad (9)$$

with μ ($0 \leq \mu \leq 1$) being the weighting factor and $\mathbf{e} = (e[0], \dots, e[n], \dots, e[N-1])^T$ is the $N \times 1$ vector of the error defined as

$$\mathbf{e} = \frac{\mathbf{y}}{G_0} - \mathbf{u} \quad (10)$$

where G_0 determines the desired linear gain of the PA, and where \mathbf{y} and \mathbf{u} are the $N \times 1$ vectors of the PA output and the transmitted input, respectively.

In Section III, we will discuss a possible FPGA implementation of the DPD adaptation considering a reduced basis of coefficients that can be estimated independently, thanks to the APCA technique.

III. INDEPENDENT DPD PARAMETER ESTIMATION BASED ON THE APCA ALGORITHM

As discussed earlier, there are several order reduction techniques published in the literature targeting a simplified implementation of the DPD function in the forward path. In this section instead, we will focus on how to address the LS matrix inversion of the covariance matrix to extract the DPD model parameters in order to enable implementation in an FPGA (containing a PL device and a processing system).

Taking into account, the transformation matrix $\hat{\mathbf{U}}$ in (8) with orthogonal basis functions, the coefficients extraction in (9) can be rewritten as

$$\hat{\mathbf{w}}[n+1] = \hat{\mathbf{w}}[n] + \mu (\hat{\mathbf{U}}^H \hat{\mathbf{U}})^{-1} \hat{\mathbf{U}}^H \mathbf{e} \quad (11)$$

where

$$\mathbf{w} = \mathbf{V}\hat{\mathbf{w}} \quad (12)$$

where by taking into account the orthogonal basis functions in $\hat{\mathbf{U}}$

$$(\hat{\mathbf{U}}^H \hat{\mathbf{U}})^{-1} = (\lambda_1^{-1}, \dots, \lambda_j^{-1}, \dots, \lambda_L^{-1}) \mathbf{I} \quad (13)$$

with λ_j being the eigenvalues of $\mathbf{U}^H \mathbf{U}$ and $\mathbf{U}\mathbf{U}^H$ (with $j = 1, \dots, L$). The coefficients can be now estimated independently

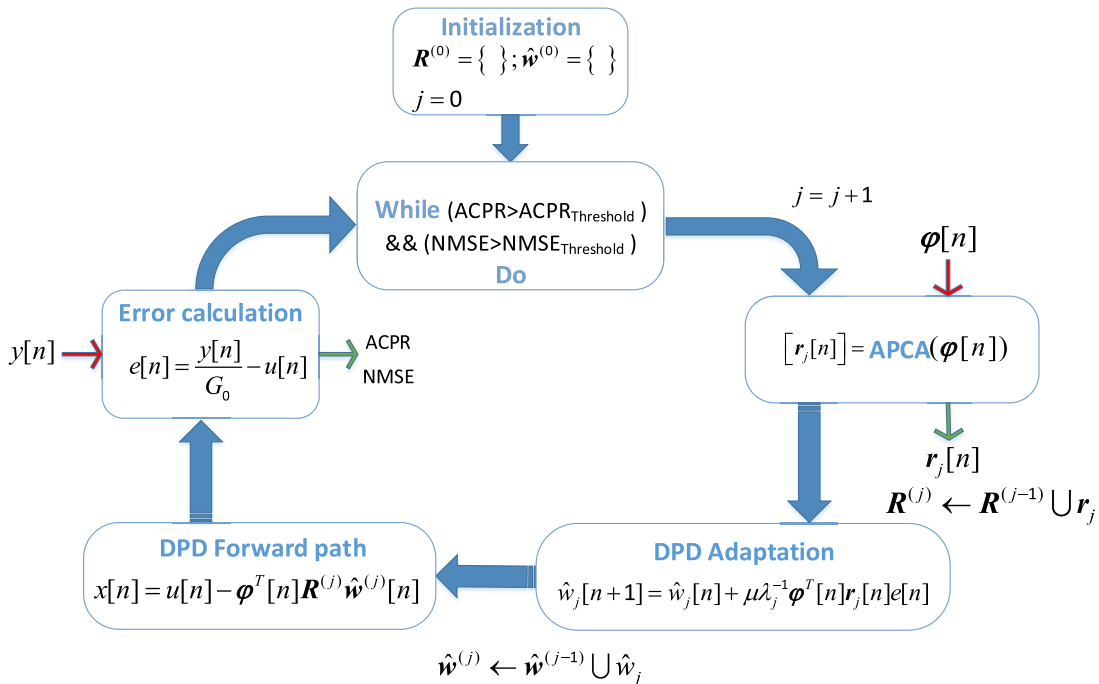


Fig. 2. Flowchart of the independent DPD identification process using APCA.

in an LMS fashion at every sample iteration n , and thus (11) becomes

$$\begin{pmatrix} \hat{w}_1[n+1] \\ \vdots \\ \hat{w}_j[n+1] \\ \vdots \\ \hat{w}_L[n+1] \end{pmatrix} = \begin{pmatrix} \hat{w}_1[n] \\ \vdots \\ \hat{w}_j[n] \\ \vdots \\ \hat{w}_L[n] \end{pmatrix} + \mu \begin{pmatrix} \lambda_1^{-1} \psi_1[n] \\ \vdots \\ \lambda_j^{-1} \psi_j[n] \\ \vdots \\ \lambda_L^{-1} \psi_L[n] \end{pmatrix} e[n]. \quad (14)$$

By exploiting the orthogonality of the resulting transformed basis functions, the coefficient adaptation can therefore be carried out independently as follows:

$$\hat{w}_j[n+1] = \hat{w}_j[n] + \mu \lambda_j^{-1} \psi_j[n] e[n] \quad (15)$$

with $j = 1, \dots, L$, where $\psi_j[n]$ is the j th transformed basis function at time n . A schematic flowchart describing the independent DPD extraction is depicted in Fig. 2. The goal is to estimate the minimum necessary number of transformed coefficients \hat{w}_j to meet the target linearity levels, specified in terms of the adjacent channel power ratio (ACPR) and normalized mean square error (NMSE). As explained in [12], with the proposed BD-APCA algorithm, the columns \mathbf{r}_j ($j = 1, 2, \dots, L$) of the transformation matrix \mathbf{R} are iteratively found one by one. Therefore, the next column is estimated by using the values of the previously extracted components. Eventually, the $M \times L$ transformation matrix $\mathbf{R} = (\mathbf{r}_1, \dots, \mathbf{r}_j, \dots, \mathbf{r}_L)$ will converge to \mathbf{V} . Therefore, as shown in Fig. 2, until the desired linearity levels are met, the algorithm increases the number of transformed coefficients to be estimated. First, after several APCA iterations, a new transformation vector \mathbf{r}_j is obtained. This vector, together with the other $j-1$ previously calculated vectors (i.e., $j-1, j-2, \dots, 1$), define

the j components of the transformation matrix $\mathbf{R}^{(j)}$. Then, each one of the j transformed coefficients $\hat{w}_j[n]$ can be estimated/updated independently (e.g., in parallel) by following an LMS approach:

$$\hat{w}_j[n+1] = \hat{w}_j[n] + \mu \lambda_j^{-1} \boldsymbol{\phi}^T[n] \mathbf{r}_j[n] e[n]. \quad (16)$$

Consequently, the input–output relationship of the DPD function in the forward path described in (3) can be rewritten as

$$x[n] = u[n] - \boldsymbol{\psi}^T[n] \hat{\mathbf{w}}[n] \quad (17)$$

or, alternatively, be expressed in terms of the original basis functions $\boldsymbol{\phi}[n]$ and the transformation matrix \mathbf{R}

$$x[n] = u[n] - \boldsymbol{\phi}^T[n] \mathbf{R} \hat{\mathbf{w}}[n]. \quad (18)$$

In Sections IV and V, we will show experimental results proving the robustness of the independent DPD algorithm described in Fig. 2.

IV. EXPERIMENTAL SETUP

The proposed independent DPD parameter extraction using the APCA technique was experimentally evaluated with a 5G-like communication system. Given the unavailability of 5G-NR waveforms at the time of testing, we employed a carrier-aggregated fast convolution filter bank multicarrier signal with 80-MHz bandwidth built with four 64-quadratic amplitude modulation modulated 20-MHz FC-FBMC channels that were carrier aggregated. Each channel follows an OFDM-like waveform configuration and a frame structure similar to that used for Long-Term Evolution (LTE) FDD (Rel. 9) and enables spectral coexistence with LTE primary systems through subcarrier group deactivation. These

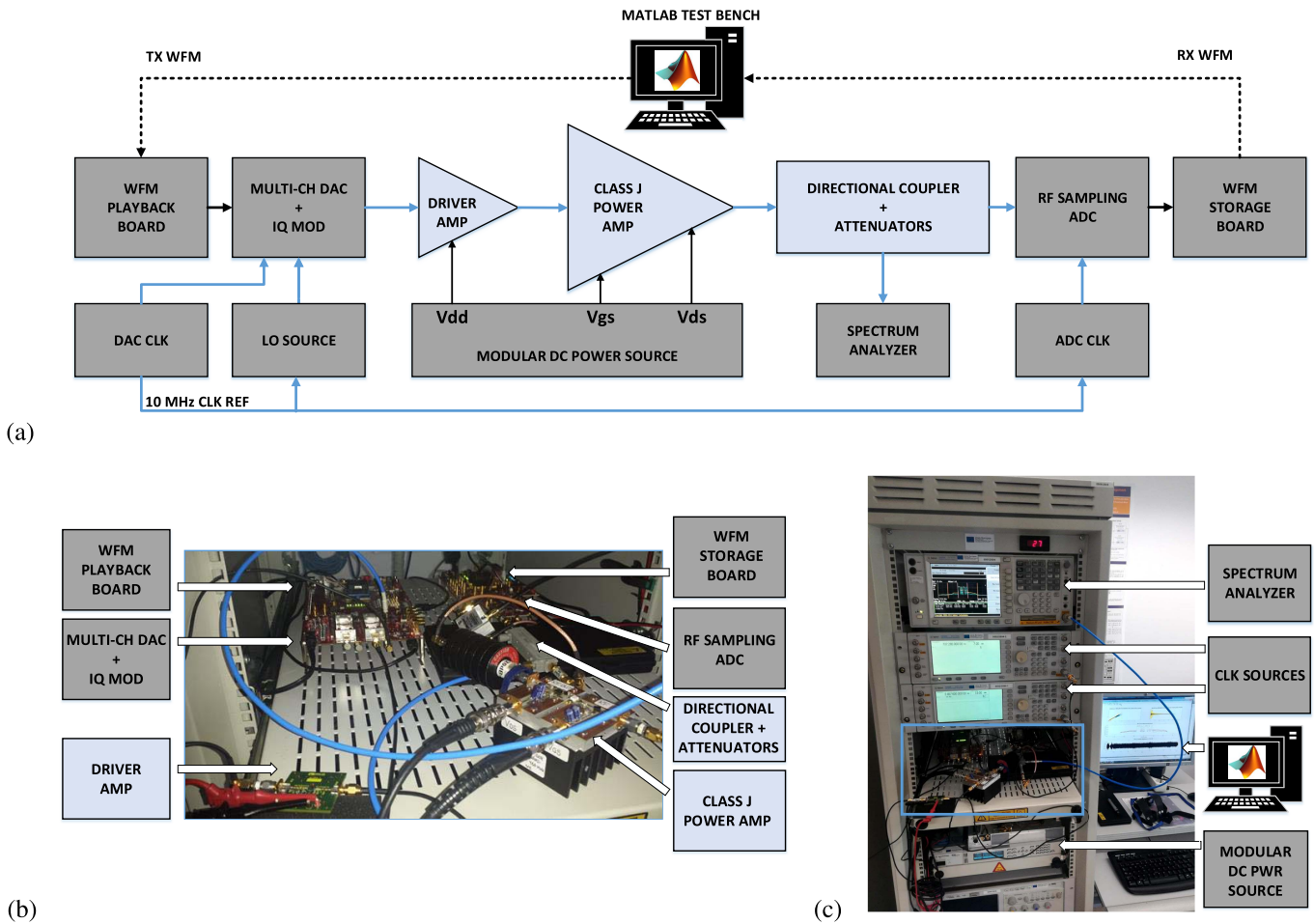


Fig. 3. (a) Block diagram of the test setup employed for experimental validation. The digital linearization test bench combines (b) waveform generation and capture boards, RF analog control parts including the PA and (c) laboratory instrumentation. TX: transmitted, RX: received, WFM: waveform, CH: channel, MOD: modulator, AMP: amplifier, ADC: analog-to-digital converter, DAC: digital-to-analog converter, CLK: clock, LO: local oscillator, dc PWR: dc power, Vdd: drain-to-drain voltage, Vgs: gate-to-source voltage, and Vds: drain-to-source voltage.

complex waveforms feature around 13-dB PAPR. In our experiments, we used a MATLAB-controlled digital linearization test bench (shown in Fig. 3) interfacing some commercial boards for waveform playback and data capture, digital-to-analog conversion (DAC), in-phase and quadrature-phase (IQ) modulator, and analog-to-digital conversion (ADC) for direct RF sampling (i.e., TI TSW1400EVM and TSW30H84EVM at Tx side and TI ADC32RF45EVM and TSW14J56EVM at Rx side). In order to account for the out-of-band distortion, a 368.64-MSa/s DPD signal with 240-MHz bandwidth was upconverted to the 875-MHz RF frequency to feed a class-J PA based on the Cree CGH35030F gallium nitride (GaN) high-electron-mobility transistor. The PA output signal (with +28-dBm mean output power) was attenuated, RF sampled at 2457.6 MSa/s, and further downsampled to the DPD signal sample rate for time alignment and DPD processing. An Agilent E4440A spectrum analyzer was used to characterize the spectrum and ACPR at the output of the PA, three Agilent E4438C signal generators were used as a data converter clock and as an IQ modulator local oscillator sources (sharing the 10-MHz reference) and an Agilent N6710B modular direct

current (dc) power system was employed to supply the active RF elements of the test setup.

V. EXPERIMENTAL RESULTS

The independent DPD closed-loop adaptation described in Fig. 2 has been validated in a MATLAB-controlled hardware test bench. Thus, instead of running the algorithm sample by sample (as it would be done in an FPGA), we considered the use of input-output data vectors for faster evaluation of the proposed algorithms. At every iteration, a different set of data was used, presenting different PAPR values (around 13 ± 1 dB) to prove the robustness of the proposed DPD linearization algorithm.

As shown in (9), typically, the Moore–Penrose inverse (i.e., $(U^H U)^{-1} U^H$) is used to solve the LS identification. However, when the covariance matrix $(U^H U)$ is ill-conditioned, the values of the estimated coefficients are no longer reliable. As discussed earlier in Section I, most of the published DPD solutions based on MATLAB designs address this problem by using MATLAB’s backslash operator (“\”). Table I shows the linearization achieved [in terms of ACPR,

TABLE I
DPD PERFORMANCE COMPARISON

Configuration	Coeff. No.	NMSE [dB]	ACPR [dBc]	EVM [%]
80 MHz FC-FBMC	No.	-18.5	-36.25	5.75
No DPD	-	-18.5	-36.25	5.75
Full DPD w/ Matlab's '\'	322	-41.1	-50.7	1.05
Indep. DPD w/ APCA	60	-38.7	-45.3	1.35

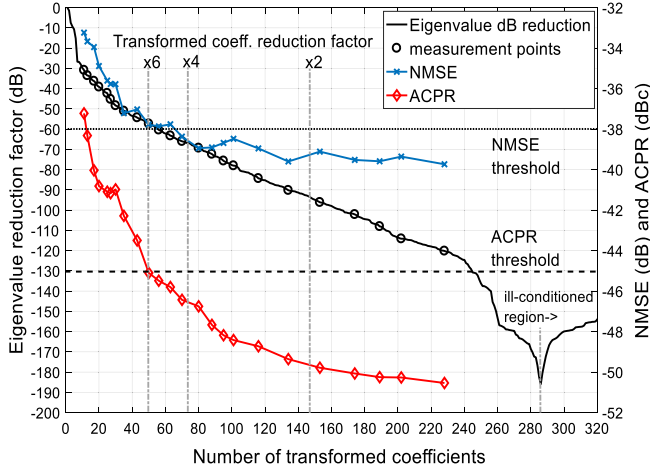


Fig. 4. NMSE and ACPR versus coefficients reduction.

NMSE, and error vector magnitude (EVM) in the worst case channel] when considering two cases: 1) the direct estimation of the best performing 322 coefficients found by brute force using MATLAB's backslash operator (since using the Moore–Penrose inverse the coefficient estimation diverges) and 2) the independent DPD adaptation of 60 coefficients using APCA.

The original data matrix containing 322 basis functions could have been reduced by applying some model order reduction such as OMP (as the authors already did in [4]). However, this is not the scope of this paper and, by assuming an initial data matrix with 322 basis functions, we can prove the regularization and coefficient reduction capabilities of the proposed approach. Having said this, and just to have some order reduction assessment, the authors have applied OMP to reduce the original data matrix in the forward path and found that by using 227 coefficients, the performance being attained is very similar to that featured by the 322 coefficients found by the brute force. Therefore, the order reduction factor in the feedback path would be around 3.5 if we consider that, to meet the target $ACPR \leq -45$ dBc and $NMSE \leq -38$ dB, only 60 coefficients of the new transformed basis were required. This number is consistent with the experiment results shown in Fig. 4 which are further detailed in the next paragraph.

Before applying APCA to iteratively build the 60 coefficients, several tests were conducted to characterize the performance of PCA versus the number of transformed coefficients. In order to have good measurement resolution and better exploit the eigenvalue information, the following procedure was used: first, the eigenvectors and eigenvalues of the matrix $U^H U$ were calculated to build the transformation

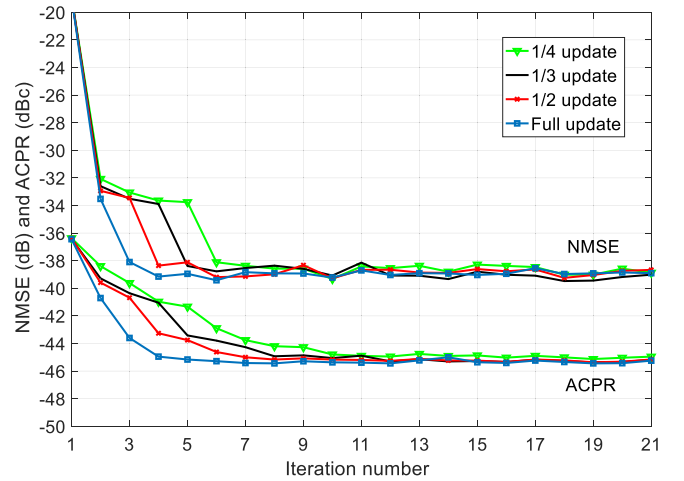


Fig. 5. Independent partial identification of DPD coefficients per iteration.

matrix for the full set of 322 coefficients. Second, the ratio in decibel between the absolute value of the last diagonal element (having the lowest value for a given number of transformed coefficients) and the absolute value of the first diagonal element (having the maximum value) was characterized in MATLAB as in Fig. 4 (black trace). Third, every measurement point defined a maximum number of consecutive transformation matrix columns (i.e., comprising a specific eigenvalue decibel reduction factor) and thus DPD coefficients that were employed in every PCA DPD experiment. A total of 25 experiments of 21 iterations each were conducted (different PAPR statistically representative waveforms were applied to check the robustness). After a training period of 10 iterations (by using a decreasing μ weighting factor), the NMSE and ACPR values for every experiment became stabilized and the worst case value was recorded and annotated in Fig. 4. Finally, this information is later used to validate the APCA operation by comparing the number of coefficients progressively obtained by APCA to reach the required performance versus that offered by the original PCA DPD approach taking the whole coefficient set from the first iteration.

By considering the 60 coefficients being required to meet the NMSE and ACPR thresholds and before experimenting with the BD-APCA method, another set of PCA experiments was set to demonstrate the independent estimation of coefficients. In these experiments, at every DPD iteration, only a portion of the overall transformed coefficients was updated, while the rest was left unaltered. The results in Fig. 5 satisfactorily show that a similar performance is reached when comparing full update with partial update at different independent update ratios. As expected, a lower number of coefficients being updated take more iterations to reach the required performance. However, by properly tuning the μ weighting factor being applied at every DPD iteration, the delay can be minimized so that the desired performance is reached during the aforementioned 10-iteration training period.

In the following, we will discuss the APCA experimental validation. Fig. 6 shows the NMSE and ACPR evolution when considering that at each iteration, a new component is

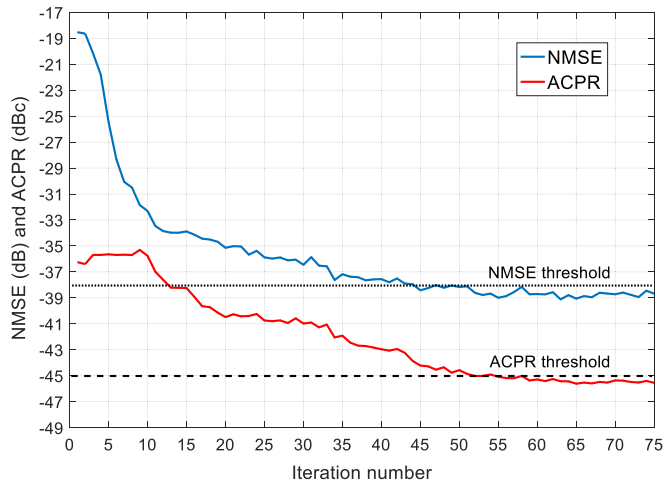


Fig. 6. Evolution of the NMSE and the ACPR considering up to 60 components and 75 iterations.

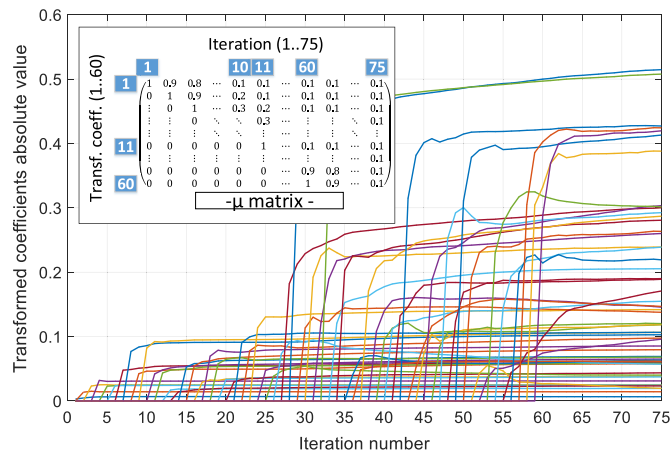


Fig. 7. Evolution of the absolute value of the 60 DPD coefficients.

included in the estimation set and thus a new coefficient is estimated. By following the algorithm described in Fig. 2, we can see that, thanks to the orthogonal property even if at every iteration, a new coefficient is included which can be estimated independently and each one contributes to the linearization. After a given number of coefficients (in our particular case 60), the targeted NMSE and ACPR are reached and then we can stop adding components and coefficients to the estimation set. In our experiment, we have considered up to 75 iterations to check the method's performance stability once the number of coefficients is no longer increased.

Fig. 7 shows the evolution of the absolute value of the estimated coefficients. The estimation is perfectly regularized, the absolute value of the coefficients is below 1, and every time a new coefficient is added as it converges in few iterations and, because of their independence, the value of the already existing coefficients is kept unaltered despite the addition of new coefficients to the estimation set. This cannot be done with the original basis, since the basis functions are not orthogonal and thus the value of the coefficients is dependent on the number of selected basis. This solution is very versatile because, thanks to the independent estimation, several strategies to reduce the

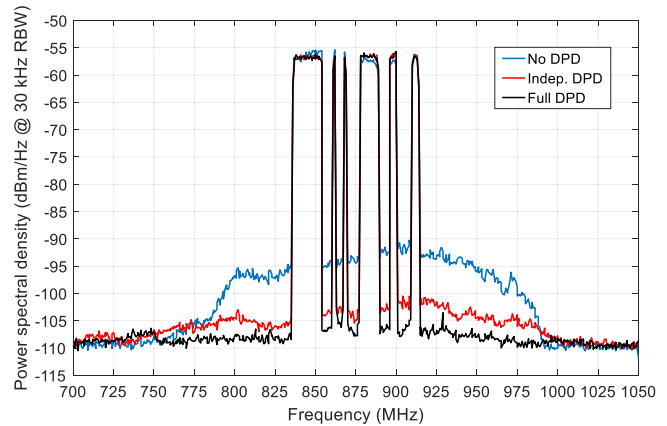


Fig. 8. Spectra of the PA output before and after DPD linearization.

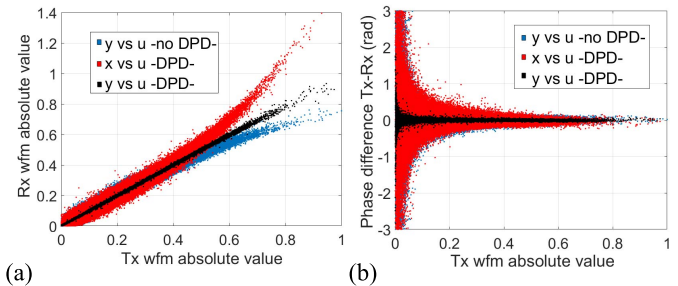


Fig. 9. (a) AM-AM and (b) AM-PM characteristics before and after DPD linearization.

computational complexity can be considered. For example, we could adapt only one coefficient at every iteration following a round robin scheduling and thus leaving the others unadapted until their turn arrives. Fig. 7 also depicts the μ DPD weighting factor strategy being applied at every experiment iteration. This factor takes into account the first iteration at which such coefficient was generated and the following updates being applied to that coefficient as if there was a specifically bounded training period for it. The rationale behind such a strategy is to contribute to guarantee the convergence time and to a shorter extent to help preserving the regularization.

Finally, Figs. 8 and 9 show the unlinearized and linearized spectra taken from the spectrum analyzer and the amplitude modulation-to-amplitude modulation (AM-AM) and amplitude modulation-to-phase modulation (AM-PM) characteristics, respectively, when considering independent DPD adaptation with 60 coefficients.

VI. CONCLUSION

This paper has shown the versatility of the independent DPD identification method using the PCA theory to find a transformed matrix with new orthogonal components. With the independent DPD identification, it is possible to tradeoff updating or identification convergence time versus computational complexity. That is, updating in parallel at every iteration (in a more relaxed time scale than the real-time digital predistorter in the forward path), following an LMS gradient technique, as many components as you want (tradeoff convergence time vs. FPGA resources). The independent DPD

relies on the need to find a transformation matrix, which thanks to the APCA algorithm in [12] can be iteratively found and, moreover, it is suitable to be implemented online in an FPGA. Therefore, in this paper, we proved a new identification algorithm by merging the APCA technique with the closed-loop DPD adaptive estimation. This way, we obtain a robust independent DPD identification by using the minimum necessary coefficients in the adaptation subsystem to meet specific linearity levels.

APPENDIX BD-APCA ALGORITHM

Input: the $M \times 1$ vector containing the basis functions $\boldsymbol{\varphi}^T[n] = (\varphi_1[n], \dots, \varphi_i[n], \dots, \varphi_M[n])$.

Output: The transformation matrix $\mathbf{R} = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_L)$.

1) Set $j = 1$, initialize small random values to the matrix \mathbf{R} and assign $\mathbf{b}_j[n] = \boldsymbol{\varphi}^*[n]$, with $n = 0, 1, \dots, N - 1$, where $*$ denotes the complex conjugate.

2) Set $k = 1$, $\mathbf{B}_j = (\mathbf{b}_j[0], \mathbf{b}_j[1], \dots, \mathbf{b}_j[N - 1])^T$.

2.1) For $n = 0, 1, \dots, N - 1$, update the vector \mathbf{r}_j as follows:

$$\begin{aligned} \mathbf{r}_{j,k}[n+1] \\ = \mathbf{r}_{j,k}[n] + \eta_{j,k}(d_{j,k}[n])^* \mathbf{b}_j[n] - d_{j,k}[n] \mathbf{r}_{j,k}[n] \end{aligned} \quad (19)$$

where $d_{j,k}[n] = \mathbf{r}_{j,k}^H[n] \mathbf{b}_j[n]$ and the learning rate parameter is

$$\eta_{j,k} = \sigma \cdot \text{trace}(\mathbf{B}_j^H \mathbf{B}_j) / k \quad (20)$$

with σ being a constant factor. The trace can be calculated as the sum of the vector's squared modulus, thus avoiding the need to calculate the full correlation matrix.

2.2) Increment k by 1, go back to step 2.1, and continue until the values of $\mathbf{r}_{j,k}$ become steady. Store the estimated eigenvector $\mathbf{r}_j = \mathbf{r}_{j,k}[N]$. Go to step 3.

3) Deflat the data vectors $\mathbf{b}_{j+1}[n]$, with $n = 0, 1, \dots, N - 1$

$$\mathbf{b}_{j+1}[n] = \mathbf{b}_j[n] - (\mathbf{r}_j^H \mathbf{b}_j[n]) \mathbf{r}_j. \quad (21)$$

4) Increment j by 1 and go back to step 2, continue until the last \mathbf{r}_j is obtained.

ACKNOWLEDGMENT

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

REFERENCES

- [1] N. Mrabet, I. Mohammad, F. Mkaem, C. Rebai, and S. Boumaiza, "Optimized hardware for polynomial digital predistortion system implementation," in *Proc. IEEE Top. Conf. Power Amplif. Wireless Radio Appl. (PAWR)*, Jan. 2012, pp. 83–84.
- [2] A. Molina, K. Rajamani, and K. Azadet, "Digital predistortion using lookup tables with linear interpolation and extrapolation: Direct least squares coefficient adaptation," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 3, pp. 980–987, Mar. 2017.

- [3] 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 Trans. Microw. Theory Techn.*, vol. 56, no. 2, pp. 372–384, Feb. 2008.
- [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. IEEE Top. Conf. RF/Microw. Power Amplif. Radio Wireless Appl. (PAWR)*, Jan. 2018, pp. 47–50.
- [5] R. N. Braithwaite, "Digital predistortion of an RF power amplifier using a reduced volterra series model with a memory polynomial estimator," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 10, pp. 3613–3623, Oct. 2017.
- [6] D. Wisell, J. Jaldén, and P. Handel, "Behavioral power amplifier modeling using the LASSO," in *Proc. IEEE Instrum. Meas. Technol. Conf. (IMTC)*, May 2008, pp. 1864–1867.
- [7] 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.
- [8] J. Peng, S. He, B. Wang, Z. Dai, and J. Pang, "Digital predistortion for power amplifier based on sparse Bayesian learning," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 63, no. 9, pp. 828–832, Sep. 2016.
- [9] J. Reina-Tosina, M. Allegue-Martínez, C. Crespo-Cadenas, C. Yu, and S. Cruces, "Behavioral modeling and predistortion of power amplifiers under sparsity hypothesis," *IEEE Trans. Microw. Theory Techn.*, vol. 63, no. 2, pp. 745–753, Feb. 2015.
- [10] S. D. Muruganathan and A. B. Sesay, "A QRD-RLS-based predistortion scheme for high-power amplifier linearization," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 10, pp. 1108–1112, Oct. 2006.
- [11] Z. Wang, W. Chen, G. Su, F. M. Ghannouchi, Z. Feng, and Y. Liu, "Low computational complexity digital predistortion based on direct learning with covariance matrix," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 11, pp. 4274–4284, Nov. 2017.
- [12] 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 *IEEE MTT-S Int. Microw. Symp. Dig.*, Jun. 2018, pp. 160–163.
- [13] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2009.
- [14] Y. Zhang and Y. Ma, "CGHA for principal component extraction in the complex domain," *IEEE Trans. Neural Netw.*, vol. 8, no. 5, pp. 1031–1036, Sep. 1997.
- [15] P. L. Gilabert *et al.*, "Order reduction of wideband digital predistorters using principal component analysis," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Jun. 2013, pp. 1–7.
- [16] R. N. Braithwaite, "A comparison of indirect learning and closed loop estimators used in digital predistortion of power amplifiers," in *IEEE MTT-S Int. Microw. Symp. Dig.*, May 2015, pp. 1–4.
- [17] D. R. Morgan, Z. Ma, J. Kim, M. G. Zierdt, and J. Pastalan, "A generalized memory polynomial model for digital predistortion of RF power amplifiers," *IEEE Trans. Signal Process.*, vol. 54, no. 10, pp. 3852–3860, Oct. 2006.
- [18] R. N. Braithwaite, "General principles and design overview of digital predistortion," in *Digital Front-End in Wireless Communications and Broadcasting*, F. Luo, Ed. Cambridge, U.K.: Cambridge Univ. Press, 2011, ch. 6, pp. 143–191.



David López-Bueno (M'13) received the M.Sc. degree in telecommunication engineering from the Universitat Politècnica de Catalunya, Barcelona Tech, Barcelona, Spain, in 2005, where he is currently pursuing the Ph.D. degree at the Department of Signal Theory and Communications.

From 2003 to 2006, he was an RF Module and Monolithic Microwave Integrated Circuit Design Engineer with space companies. In 2006, he joined the Centre Tecnològic de Telecomunicacions de Catalunya, Barcelona, as a Research Engineer. Since 2006, he has been participating in a number of research projects dealing with RF transceivers, 4G/5G multiple-input multiple-output OFDM-based laboratory demonstrators, and assessing the interference between heterogeneous systems. Since 2013, he has also been involved in projects in the field of wideband power amplifier digital linearization for radio equipment.



Quynh Anh Pham (S'18) received the M.Sc. degree in computer science and engineering from Soongsil University, Seoul, South Korea, in 2015. She is currently pursuing the Ph.D. degree at the Department of Signal Theory and Communications (TSC), Universitat Politècnica de Catalunya (UPC), Barcelona Tech, Barcelona, Spain.

Since 2017, she has been with TSC, UPC, Barcelona Tech. Her current 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 Ph.D. degree from the Universitat Politècnica de Catalunya (UPC), Barcelona Tech, Barcelona, Spain, in 1990 and 1996, respectively.

In 1991, he joined the Department of Signal Theory and Communications, UPC, Barcelona Tech, where he is currently an Associate Professor. His current research interests include adaptive control and the use of signal processing strategies for power efficiency improvement in communications systems.



Pere L. Gilabert (M'09–SM'13) received the M.Sc. degree in telecommunication engineering and Ph.D. degree (Hons.) from the Universitat Politècnica de Catalunya (UPC), Barcelona Tech, Barcelona, Spain, in 2002 and 2008, respectively. His master's thesis was with the University of Rome–La Sapienza, Rome, Italy, under an Erasmus Exchange Grant.

In 2003, he joined the Department of Signal Theory and Communications, UPC, Barcelona Tech, where he is currently an Associate Professor with the Castelldefels School of Telecommunications and

Aerospace Engineering. His current research interests include linearization techniques and digital signal processing solutions for highly efficient transmitter architectures.