Beyond the Moore–Penrose Inverse: Strategies for the Estimation of Digital Predistortion Linearization Parameters

Pere L. Gilabert, R. Neil Braithwaite, and Gabriel Montoro

Pere L. Gilabert (plgilabert@tsc.upc.edu) and Gabriel Montoro (gabriel.montoro@upc.edu) are with the Dept. of Signal Theory and Communications, Universitat Politècnica de Catalunya (UPC-Barcelona Tech.), c/ Esteve Terradas, 7, 08860, Castelldefels, Spain.

R Neil Braithwaite (neil.braithwaite@ieee.org) is with Keysight Technologies, Santa Clara, CA, USA.

I. Introduction

Digital predistortion (DPD) is the linearization technique most-often used nowadays to cope with the inherent trade-off between linearity and efficiency in power amplifiers (PA). Adaptation is needed to optimize DPD performance. This paper reviews strategies for estimating the parameters controlling the digital predistortion linearizer.

As depicted in the block diagram of Fig. 1, the DPD linearization system can be divided into two main subsystems: the DPD function (i.e., parametric mathematical model) in the forward path that is responsible for predistorting the input signal, and an adaptation subsystem in the feedback or observation path, where the parameters characterizing the nonlinear DPD function in the forward path are estimated and updated. The DPD function in the forward path has to operate in real-time in a programmable logic (PL) unit (e.g., in a system on chip (SoC) FPGA device). Consequently, the DPD function in the forward path should be designed as simply as possible to save hardware resources and meet timing constraints. The DPD coefficients can be estimated/adapted iteratively in a processing system (PS) unit on a much slower time scale than in the forward path (i.e., not in real-time), unless the PA time-variant behavior changes so fast that it requires real-time adaptation.

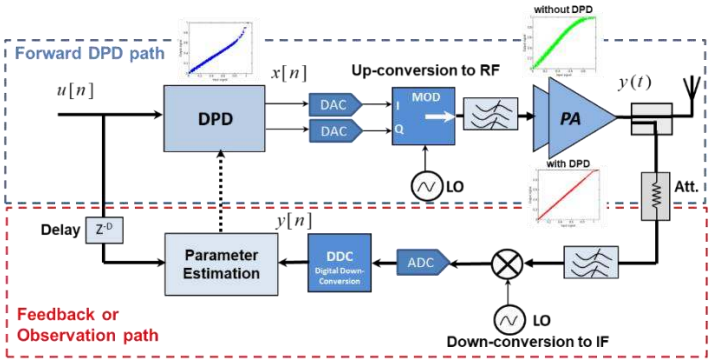

Fig. 1. Block diagram of an adaptive digital predistortion linearizer.

There is plenty of literature addressing the problem of PA and DPD behavioral modeling to characterize nonlinear and dynamic behavior when considering wide bandwidth signals [1]-[2], carrier aggregation, multiband or multi-antenna configurations [3]-[4], or taking into account high-efficiency amplification topologies (e.g., Doherty PAs, load-modulated balanced amplifiers, envelope tracking PAs, or outphasing transmitters) [5]-[6]. The DPD function in the forward path is often implemented as a weighted sum of nonlinear basis waveforms.

Some equations for a better(?) understanding, please don't panic.

The authors are aware that this is a magazine overview paper; however, the topic sometimes requires making use of equations to explain certain concepts. Even so, while the equations below are quite helpful, the reader can choose to ignore the mathematical equations and still gain a general idea of the different parameter estimation approaches described here.

Fig. 2 shows a simplified block diagram of an adaptive DPD showing both direct and indirect learning adaptation and where the mathematical formulae are modified accordingly. The output of the DPD function is defined as the signal to be transmitted plus a distortion term (which is the opposite of the estimated PA distortion) which, as shown in Fig. 2, is defined as the linear combination of nonlinear functions weighted by certain parameters [7] (i.e.,

$x[n]=u[n]+\boldsymbol{\varphi}^T[n]\mathbf{w}$), with $\boldsymbol{\varphi}[n]$ being the *Mx1* (column) vector containing the nonlinear basis functions of a specific behavioral model, and $\mathbf{w}$ being the *Mx1* vector of coefficients (weights, parameters). As shown in Fig. 2, in adaptive DPD, the DPD coefficients (e.g., initially set to zero $\mathbf{w}^{(0)}=0$) are found following an iterative approach, where the new vector of coefficients is obtained from the previous one minus an increment (estimated coefficient error):

$$\mathbf{w}^{(k+1)}=\mathbf{w}^{(k)}-\mu_k\,\Delta\mathbf{w}^{(k)} \tag{1}$$

and reaching a steady state after several (*k*) iterations ($\Delta\mathbf{w}\approx 0$), at which point the system is said to be converged. Following a statistical approach, we can apply the minimum mean square error (MMSE) criterion to estimate the optimum increment of the DPD coefficients. The cost function is then defined as the mean square error $J(\mathbf{w})=E\left\{\left|e[n]\right|^2\right\}$, where $E\{\cdot\}$ is the mathematical expectation and the error, focusing on the direct learning (or closed loop) approach in Fig. 2, is defined as $e[n]=\left(y[n]/G_0\right)-u[n]$, with G_0 being the linear gain. In the method of the steepest descent, the successive adjustments applied to the weight vector $\mathbf{w}$ are in a direction opposite to the gradient vector of the cost function, $\nabla J(\mathbf{w})$. Finding the optimum parameters (in a statistical sense) requires solving the Wiener–Hopf equation, and thus we need to calculate the auto-correlation matrix $E\left\{\boldsymbol{\varphi}[n]\boldsymbol{\varphi}^H[n]\right\}$ and the cross-correlation vector $E\left\{\boldsymbol{\varphi}[n]e[n]\right\}$. However, the statistical information contained in the auto-correlation matrix and cross-correlation vector may not be available. As an alternative, one of the most common approaches found in the literature to extract the DPD parameters is the method of least squares (LS), based on a batch-processing approach that relies on processing blocks of nonstationary data. The LS solution will converge to the MMSE estimation [12]. For doing so we extend the basis function vector taking into account several data observations (i.e., n=0,1,…,*N*-1 samples), obtaining the *N*x*M* data matrix $\boldsymbol{\Phi}$. Therefore, the best fit for the $\mathbf{w}$ parameters is obtained by minimizing the sum of the squares of the error : $\min_{\mathbf{w}}\sum_{n=0}^{N-1}\left|e[n]\right|^2$. Again, with the direct learning approach we are forcing the PA output to be equal to the original signal to be transmitted except for the PA linear gain G_0. Assuming zero mean basis functions from now on, let us define the covariance matrix (equivalent to the correlation matrix if a normalization factor is applied to the data) as follows $\mathbf{Q}=\mathrm{cov}(\boldsymbol{\Phi})=\frac{1}{N-1}\left(\left(\boldsymbol{\Phi}-E\{\boldsymbol{\Phi}\}\right)^H\left(\boldsymbol{\Phi}-E\{\boldsymbol{\Phi}\}\right)\right)\approx\boldsymbol{\Phi}^H\boldsymbol{\Phi}$, while the covariance vector is $\mathbf{r}=\boldsymbol{\Phi}^H\mathbf{e}$. Therefore, the LS solution to this system, $\Delta\mathbf{w}=\left(\boldsymbol{\Phi}^H\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^H\mathbf{e}=\mathbf{Q}^{-1}\mathbf{r}$, requires the calculation of the Moore–Penrose inverse (as depicted in Fig. 2) and lacks a unique solution when processing blocks of nonstationary data.

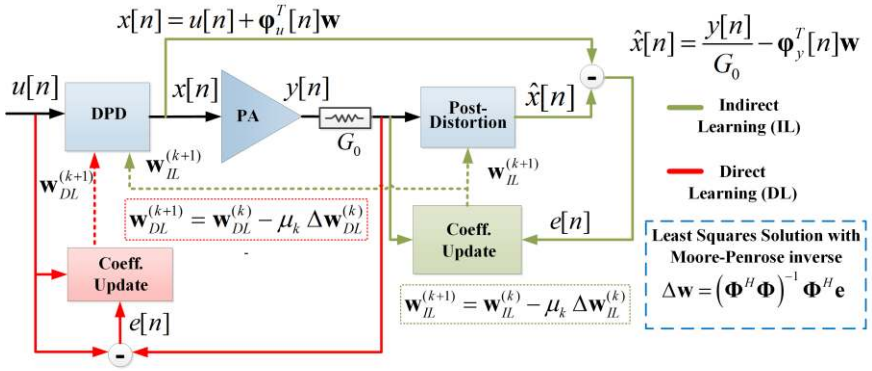

Fig. 2. General formulation of an adaptive DPD with direct or indirect learning.

II. It's the algebra, stupid.

Applying DPD to compensate for the nonlinear behavior of a power amplifier or other device requires an understanding of mathematical principles and how they are used to represent the behavior of a physical system. Novices in the field often apply linear algebra, such as LS, as written in prior papers, and don't get the results they expected. They assume that there are shortcomings in the computerized computation, such as precision problems in a matrix inverse. A common approach to inverting LS singular matrices is to use MATLAB's backslash operator "\" (otherwise known as the "*mldivide*" function) that employs a QR solver for dense non-square matrices like those typically found in DPD. In reality, the problem is that linear algebra can only be used to approximate a system locally about a specific operating point when the system is nonlinear and the nonlinearity is weak. Stronger nonlinearities, such as due to clipping or saturation, invalidate this assumption, causing the math to fail to produce useful results. Modifications are needed.

So why does algebra fail? The first problem is the computation of the inverse of the covariance matrix $\mathbf{Q}$. If the basis waveforms are highly correlated, the covariance matrix $\mathbf{Q}$ becomes close to singular (ill-conditioned). Consequently, the LS estimate tends to be highly sensitive to random errors in the response ($y[n]$), such as random noise or quantization noise of the measurement setup. The risk is that the update process for the DPD coefficients will drift along the homogeneous mode, possibly saturating the coefficients digitally (a homogeneous mode is a non-zero vector $\mathbf{h}$ where $J(\mathbf{w}) \approx J(\mathbf{w}+\mathbf{h})$). Regularization is used to compute a value of $\mathbf{Q}^{-1}$ that suppresses homogeneous modes so that the solution for $\mathbf{w}$ is unique.

The idea behind regularization is to add a penalty term to the cost function to prevent unwanted solutions [8]-[9]. One of most commonly used regularization methods is the ridge regression or Tikhonov regularization [8], where the least squares regression cost function is subject to a constraint on the squared Euclidean norm of the vector of coefficients. Therefore, the goal is to minimize the residual sum of squares (RSS) subject to a constraint on the sum of squares of the coefficients. This constraint forces the coefficients to stay within a sphere of radius *t*, as shown in Fig. 3.

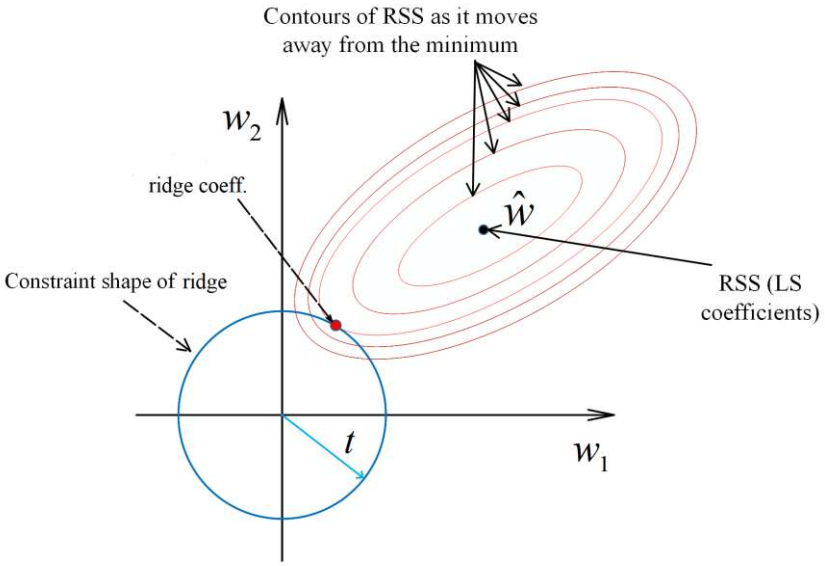

Fig. 3. Graphical description of the ridge regression or Tikhonov regularization.

Regularization can be understood by considering the relationship between the estimated coefficient error $\Delta\mathbf{w}$ and the actual coefficient error $\delta\mathbf{w}$. The prediction of the vector $\mathbf{r}$ using the actual coefficient error is $\hat{\mathbf{r}} = \mathbf{Q}\delta\mathbf{w}$.

The prediction lacks the noise and quantization present in the measured $\mathbf{r}$. The matrix $\mathbf{Q}$ provides a set of gains defined by its eigenvalues that follow the corresponding eigenvectors. If $\delta\mathbf{w}$ is decomposed into modes following the eigenvectors of $\mathbf{Q}$, each mode is amplified by the respective eigenvalue of $\mathbf{Q}$. The highest eigenvalues of $\mathbf{Q}$ are considered dominant because small changes in $\delta\mathbf{w}$ along the corresponding eigenvectors cause large changes in $\mathbf{r}$, the latter of which is a function of the output waveform error $e[n]$. In contrast, small changes in $\delta\mathbf{w}$ along non-dominant eigenvectors (lowest eigenvalues of $\mathbf{Q}$) have minimal effect on $e[n]$, which means that such modes can be suppressed without significantly degrading the DPD performance.

The LS estimation is based on the reverse relationship, $\Delta\mathbf{w} = \mathbf{Q}^{-1}\mathbf{r}$. The eigenvalues of $\mathbf{Q}^{-1}$ are the inverse of those in $\mathbf{Q}$; however, the eigenvectors are the same. This means that the reverse gain is high for the modes (eigenvectors) associated with the smallest eigenvalues of $\mathbf{Q}$, amplifying noise and quantization present in $\mathbf{r}$. A pseudo-inverse $\mathbf{S}$ (e.g., in ridge regression $\mathbf{S} = \left(\mathbf{Q} + \lambda_{ridge}\mathbf{I}\right)^{-1}$ where λ_{ridge} is the shrinkage parameter) is used in place of $\mathbf{Q}^{-1}$ to limit the maximum reverse gain or to suppress undesirable modes of $\mathbf{r}$. Thus, if we equate the measured and predicted $\mathbf{r}$ because the undesirable modes are suppressed, the relationship between the estimated and actual coefficient errors is $\Delta\mathbf{w} = \mathbf{S}\mathbf{r} = \mathbf{S}\mathbf{Q}\delta\mathbf{w}$. This makes the LS estimate more robust because $\mathbf{SQ}$ has unity gain for the dominant eigenvectors of $\mathbf{Q}$ and low gains for the noise-sensitive eigenvectors. $\mathbf{S}$ will have the same eigenvectors as $\mathbf{Q}$ if ridge regression or a truncated singular value decomposition (unwanted eigenvalues in $\mathbf{S}$ are set to zero) is used. For approaches involving feature extraction (section III.3), the reduced subset of the original parameters has an effect similar to making the eigenvalues of $\mathbf{S}$ equal to zero for the noise-sensitive eigenvectors.

The second cause of failure is that the update process is not exact. The coefficient error $\Delta\mathbf{w}$ models the residual error of the combined DPD and PA. We are using it as an estimate of the DPD coefficient error. The two are equivalent only when the PA is linear. When the PA saturates, or clipping occurs in the digital portion prior to the PA, the update fails. The estimate of $\mathbf{w}$ shows that the saturated peaks of $y[n]$ are too low compared to $G_0 \cdot u[n]$, but no adjustment $\Delta\mathbf{w}$ will increase the clipped peaks further. Repeated updates will cause the adaptive process to diverge and go unstable. This problem needs to be addressed by design or by penalizing excessive peaking in the predistorted signal $x[n]$. The error minimization approaches described in section III.2 are robust to the effect of strong nonlinearities. The other approaches described in this paper are not, in the form presented, requiring the peak power to be backed off from saturation using crest factor reduction of the input signal and/or a reduced average power to provide headroom for the predistortion to be effective.

The least squares (LS) approach with regularization can be viewed as the baseline approach to coefficient estimation. It computes coefficient updates using data captured in batches of N samples. Regularization solves the first problem of ill-conditioning of Q due to correlation between the waveforms in the basis set. The second problem of poor performance when the PA saturates is not addressed as part of the estimator.

In the following we will discuss some of the identification approaches proposed in the literature to extract the DPD parameters and try to circumvent the causes of failure. The overview is then organized as follows.

Section III.1 is a review of approaches that update the coefficients after each sample is captured, often referred to as real-time adaptation. The per sample methods suffer from problem 1 because they are difficult to regularize and they do not solve the second problem caused by PA saturation either. It might be argued that the per sample methods are better suited for FPGA implementation. Their inclusion in this paper is for completeness and should not be viewed as a recommendation.

Section III.2 introduces error minimization approaches whose origins could be attributed to the field of nonlinear optimization. They can be used successfully as coefficient estimators for a PA operating near saturation, solving problem 2. Some of the approaches can incorporate a dimension reduction, which solves problem 1 of ill-conditioning. Each error power minimization approach uses a batch of *N* samples to estimate the squared error so that the cost is a function of the coefficient setting rather than the input signal envelope. It is different than the LS approach because the coefficient setting is first changed using an exploratory step, and then the change in the error power is measured. There is no correlation between the error waveform and the available basis waveforms in an attempt to estimate the coefficient error, as in the LS case. The problem with the error power minimization approach is that it is slow to converge compared to the LS approach.

Section III.3 discusses feature extraction approaches that are popular within the machine learning and artificial intelligence communities. They are similar to the LS approach in that data captures of length N samples are used. However, the basis set in the estimator is transformed so that the new basis waveforms are orthogonal to each other. That is, the off-diagonal elements of the transformed matrix $\hat{\mathbf{Q}}$ are zero. Computation of the pseudo-inverse $\mathbf{S} = \hat{\mathbf{Q}}^{-1}$ is simple because it is just the inverse of the individual diagonal elements. Dimension reduction is trivial in the sense that one only has to retain the transformed basis waveforms that have a large value within the corresponding diagonal elements of $\hat{\mathbf{Q}}$, which solves problem 1. The second problem of PA saturation is not addressed as part of the estimation.

Therefore, we will discuss stochastic gradient techniques, updating coefficients sample-by-sample oriented at real-time identification, error power minimization approaches and techniques used in machine learning to extract a reduced subset of the original parameters. Fig. 4 summarizes some of the most commonly reported methods for extracting the parameters of the DPD function.

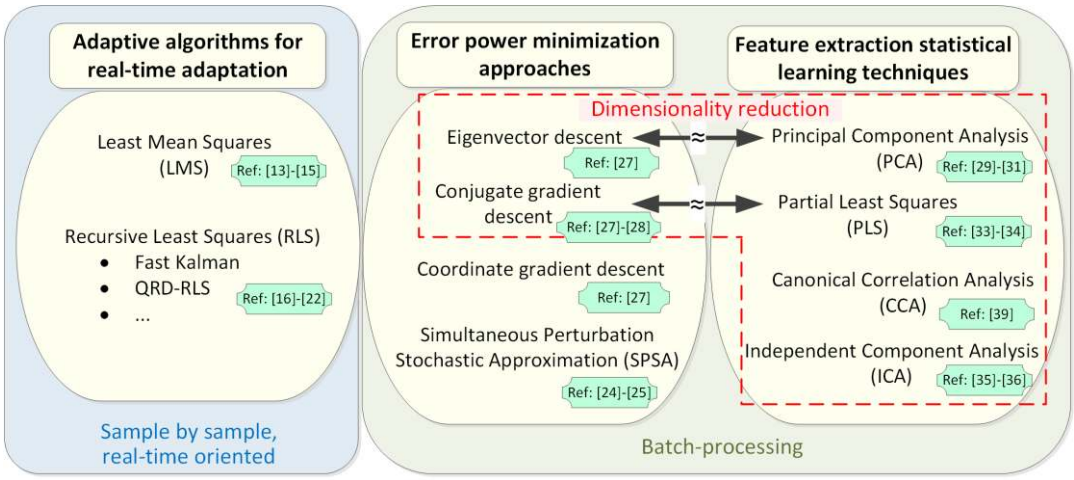

Fig. 4. Identification methods used for the extraction of DPD parameters.

III. Solutions for estimating the coefficients of an adaptive DPD linearizer

III.1 Adaptive algorithms for real-time adaptation

Adaptive filtering algorithms such as **least mean squares (LMS)** are used in DPD for real-time adaptation because of their simplicity [13]-[15]. LMS is a stochastic gradient descent method where the DPD coefficients are adapted in real-time by minimizing the instantaneous cost function based on a single realization of the estimation error. The principle of the LMS algorithm is depicted in Fig. 5: if the cost function derivative with respect to some parameter of $\mathbf{w}$ is positive, then the corresponding parameter should decrease. That is, the evolution of a parameter is contrary

to the sign of the corresponding derivative of the instantaneous cost function. The coefficient update, mirroring (1), is $\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \mu_{LMS}\,\boldsymbol{\varphi}[n]e[n]$.

Compared to LS-based minimization, LMS is a suboptimal solution, and, moreover, the solution may never converge [12]. The convergence (or even divergence) speed of the LMS algorithm is directly dependent on which step-size parameter μ_{LMS} is chosen. Summing up, the LMS adaptation is very popular for its low computational complexity introduced per iteration, but it suffers from slow convergence speed and potential stability issues.

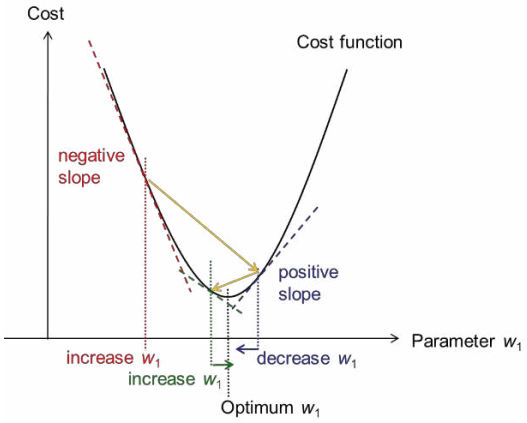

Fig. 5. Least mean squares method.

LS estimation uses batches of *N* data samples captured from the input and output signals, u[*n*] and y[*n*], for each iteration. It is possible to reformulate the estimation to be recursive and update the coefficient vector $\mathbf{w}$ after a sample is captured. The **recursive least squares (RLS) approach** has an exponential weighting ($\mu_{RLS} < 1$) where older samples are discounted. A set of recursive equations creates a matrix $\mathbf{P}$ that is roughly equal to $\mathbf{Q}^{-1}$, where $\mathbf{P}[n] = f\left\{\mathbf{P}[n-1], \boldsymbol{\varphi}[n], \mu_{RLS}\right\}$. The updated coefficient vector becomes $\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \mathbf{P}[n]\boldsymbol{\varphi}[n]e[n]$. This is not the standard form of the RLS equation set, but it does a better job of showing its relationship to the LS and LMS approaches. RLS introduces a new problem of sensitivity to quantization and saturation [22].

The convergence of the RLS is its most misrepresented property. Most researchers plot the convergence ($J(\mathbf{w})$ as a function of time) from the initial conditions, which are $\mathbf{w} = 0$ and $\mathbf{P}[0] = \sigma^2\mathbf{I}$ where σ^2 is increased to reflect the uncertainty in the initial estimate of $\mathbf{w}$. The convergence appears to be very rapid, followed by a steady state with minimal variance. It looks great. What is happening is that the RLS is making an estimate at the start using just a few samples, which produces a coefficient estimate with a high variance. However, the variance is generally less than the drop in the squared error because the initial estimate of $\mathbf{w}$ is so inexact and σ^2 is large. As more samples are accumulated, the influence of new samples drops along with the variance in the estimate of the coefficient vector. The problem with RLS is that if the system changes so that new coefficients are needed, the RLS will be unresponsive because the approach has memory. The selection of μ_{RLS} controls the trade-off between high convergence speed and low steady-state variance. Comparing the RLS and least squares approaches, RLS has a time constant proportional to $1/(1-\mu_{RLS})$ which introduces a settling time delay, whereas least squares has a buffer latency of N samples. In the end, the convergence properties of RLS and LS are similar when $\mu_{RLS} = N - 1/N$.

Let's compare the convergence properties of the RLS and LMS approaches. The general form of the per-sample coefficient update equation is $\mathbf{w}^{(k)} = \mathbf{w}^{(k-1)} - \mathbf{S}\boldsymbol{\varphi}[n]e[n]$ where $\mathbf{S} = \mathbf{P} \approx \mathbf{Q}^{-1}$ for RLS and $\mathbf{S} = \mu_{LMS}\mathbf{I}$ for LMS. Convergence depends on the condition number (ratio of the largest and smallest eigenvalues) of matrix product $\mathbf{SQ}$ where a smaller number is more favorable. What this means is that RLS will have significantly better convergence

properties than LMS when $\mathbf{Q}$ is ill-conditioned, which is often the case for basis sets using polynomial expansions. However, RLS suffers from challenges associated with the estimation of $\mathbf{P}$ as opposed to the simplicity of selecting μ_{LMS} for LMS.

Targeting a robust FPGA implementation, **QR-decomposition-based recursive least squares (QRD-RLS)** demonstrates better capability of processing incoming signals than RLS alone. When the condition number of the covariance matrix is high, then RLS becomes numerically unstable and unable to track the incoming signals. QRD-RLS is numerically more stable than RLS with a large condition number and is more able to track the incoming signals [12]. Examples of the application of a QRD-RLS algorithm for estimating the coefficients of adaptive DPD linearizers can be found in [19]-[20]. The robustness occurs because the QRD-RLS solves a simultaneous equation instead of inverting $\mathbf{Q}$.

Alternatively, the **fast Kalman** algorithm is a type of RLS algorithm proposed in [18] to take advantage of the original Kalman/Godard adaptation algorithm while reducing its original computational complexity from an order of M^2 operations per iteration to only M operations per iteration, and thus on the order of the LMS algorithm [18]. By exploiting the matrix inversion lemma to compute the Moore–Penrose inverse recursively, it is possible to avoid computing and storing the *MxM* covariance matrix. The fast Kalman algorithm is used for PA behavioral modeling and DPD purposes in [16]-[17]. As discussed above, RLS-type algorithms such as fast Kalman show an adaptation rate that is (typically an order of magnitude) faster than that of the simple LMS algorithm.

III.2 The error power minimization approach

It is possible to reformulate the problem as an error power minimization [23]. The cost J for DPD coefficient $\mathbf{w}$ is denoted by $J(\mathbf{w}) = \sum_{n=0}^{N-1} |e[n]|^2$ where $e[n]$ is the error signal for the coefficient setting $\mathbf{w}$. The time average of N samples limits the measurement variance. The collection of costs $J(\mathbf{w})$ spanning the range of coefficient settings $\mathbf{w}$ is referred to as an error surface. The goal is to find the coefficient setting that corresponds to the minimum of the error surface.

An error power minimization measures J for several coefficient settings while searching for the minimum using trial and error. What characterizes this class of approaches is that the coefficient vector $\mathbf{w}$ is intentionally disturbed (detuned) while making successive measurements of J. The drawback is that the performance of the DPD is degraded during the search, which can be seen within the output spectrum by the rising and falling of out-of-band intermodulation. The benefit is that such searches are robust; that is, less likely to diverge.

The selection of $\mathbf{w}$ over time determines the rate of convergence. A search begins by measuring the cost $J(\mathbf{w}^{(0)})$ where $\mathbf{w}^{(0)}$ is the initial coefficient setting. One could select the next setting $\mathbf{w}^{(1)}$ at random, compare $J(\mathbf{w}^{(1)})$ to $J(\mathbf{w}^{(0)})$, and retain the setting that produces the smallest value of $J(\mathbf{w})$. Repeating this process would reduce the cost over time, but convergence to the minimum of the error surface would be slow. When the error surface has favorable properties, like being convex (discussed below in this section), better search methods are available. This section considers only approaches where a descent direction is selected to define a line in a multi-dimensional space, then the selected values of $\mathbf{w}$ are constrained to the line. Once the minimum $J(\mathbf{w})$ on the line is found, a new descent direction is selected and a new search is made. This is repeated until the minimum of the multi-dimensional error surface is found. Approaches described in this section differ by the strategy used to select successive descent directions.

Let's look at the search in more detail. A descent direction vector $\boldsymbol{\delta}$ and an exploratory step size α are selected and $J(\mathbf{w}+\alpha\boldsymbol{\delta})$ is compared to $J(\mathbf{w})$. The lower of the two settings is selected and the difference is used to select the next step in order to find the minimum cost along the line $J(\mathbf{w}+\alpha_{\min}\boldsymbol{\delta})$. This is referred to as a line search. In some approaches, three collinear settings are measured, $J(\mathbf{w}-\alpha\boldsymbol{\delta})$, $J(\mathbf{w})$, and $J(\mathbf{w}+\alpha\boldsymbol{\delta})$, then a parabolic fit is used

to estimate $\alpha_{\min}$. The coefficient setting $\mathbf{w}_{\min} = \mathbf{w} + \alpha_{\min}\boldsymbol{\delta}$ is referred to as a local minimum because it results in the minimum cost J of a line search along $\boldsymbol{\delta}$. Once a local minimum is reached, the direction $\boldsymbol{\delta}$ is changed. To find the global minimum, the directions searched must span the *M*-dimensional space. 2*M* line searches may be needed when the coefficients are complex. Fewer directions need to be searched if a subset of the available directions produces a low enough J to pass the specifications.

So why would anyone use an error power minimization when the least squares approach is available? The first case is when the basis waveforms $\boldsymbol{\varphi}[n]$ are not accurately known. This is often the case when the predistortion is performed in the analog domain, but the estimation is done digitally. The other case is when the PA is operating near saturation. Because the search is based on J, it accounts for the effects of clipping. As a result, a successful search for the global minimum is possible using the error power minimization approaches under conditions that would cause LS or RLS to fail.

Descent-based searches work best when the error surface is convex. A convex surface has a bowl shape with a global minimum at a single coefficient setting denoted by $\mathbf{w}_{opt}$. The magnitude of the derivative of the cost along a line, $\left|\partial J(\mathbf{w}_{opt} + \alpha\boldsymbol{\delta})/\partial\alpha\right|$, increases with $|\alpha|$. A lesser condition is that the error surface be monotonic, where the cost $J(\mathbf{w}_{opt} + \alpha\boldsymbol{\delta})$ increases with $|\alpha|$. In general, the error surface is convex over a region where the PA operates but may be only monotonic in regions near PA saturation. The search process should be constrained to avoid entering non-monotonic regions. Note that the least squares solution assumes that the error surface is quadratic, which is a specific type of convex surface where the cost $J(\mathbf{w}_{opt} + \alpha\boldsymbol{\delta})$ increases with the square of α. More precise definitions of the convex, quadratic, and monotonic conditions for cost curves following a linear cross-section of the error surface appear in Fig. 6.

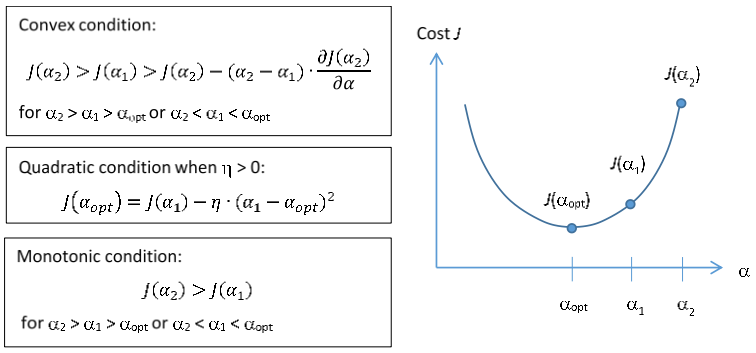

Fig. 6. Evolution cost $J(\mathbf{w} + \alpha\boldsymbol{\delta}) = J(\alpha)$ along a line on the error surface. Conditions for convex, quadratic, and monotonic curves are included.

The different error power minimization searches are characterized by the strategy used to pick the direction $\boldsymbol{\delta}$. It is preferable to use $\mathbf{Q}$ and $\mathbf{r}$ to guide the selection of the direction $\boldsymbol{\delta}$, reserving the detuning of $\mathbf{w}$ for the line search. Ideally, we would like to pick a descent direction $\boldsymbol{\delta}$ where the current setting $\mathbf{w}$ and the global minimum $\mathbf{w}_{opt}$ are on the same line. This can be done by selecting $\boldsymbol{\delta} = -\mathbf{S}\mathbf{r}$ where $\mathbf{S} \approx \mathbf{Q}^{-1}$.

Below we review two types of error power minimization methods in common use. In the first approach, the selection of the direction $\boldsymbol{\delta}$ is independent of past values of $\boldsymbol{\delta}$. These include the **gradient descent** and the **simultaneous perturbation stochastic approximation (SPSA)** [23]-[26]. The second approach uses a global cycle of *M* directions that are orthogonal in some sense and performed as 2*M* line searches (*M* directions for each of the real and imaginary components). These include the coordinate descent [27], the conjugate gradient [28], and the eigenvector

descent. Examples of the descent trajectory, which illustrate the sequence of the directions selected for the gradient descent, coordinate descent, conjugate gradient, and eigenvector descent are shown graphically in Fig. 7.

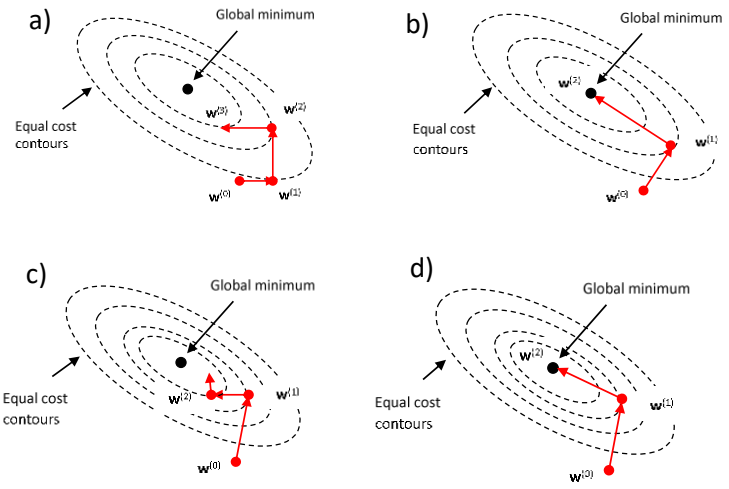

Fig. 7. The descent trajectory, or sequence of directions selected, for different error minimization methods. a) Coordinate descent, b) Eigenvector descent, c) Gradient descent, d) Conjugate gradient.

The **simultaneous perturbation stochastic approximation (SPSA) method** selects the direction using a perturbation vector $\boldsymbol{\delta}$ whose elements are either 1 or -1, selected at random. There is no requirement that the number of 1's and -1's be equal in the vector, only that the probability of 1 or -1 for each element be equal. The perturbation vector elements are 1+j, 1-j, -1+j, and -1-j for complex coefficients. Three measurements, $J(\mathbf{w}-\alpha\boldsymbol{\delta})$, $J(\mathbf{w})$, and $J(\mathbf{w}+\alpha\boldsymbol{\delta})$, are obtained, from which the desired step size $\alpha_{\min}$ is estimated using a parabolic fit. This method is useful when the gradient information from $\mathbf{r}$ is not available. The computational cost per iteration is very low; however, it will take significantly more iterations to converge than the gradient descent discussed next.

The **gradient descent** uses the gradient of the cost as the direction: $\boldsymbol{\delta}=\nabla J(\mathbf{w})$. When the error surface is quadratic, $\boldsymbol{\delta}=-\mathbf{r}$**.** It is also a useful approximation for convex and monotonic surfaces because the step size is computed using a line search. The gradient method suffers from a problem similar to LMS in that the modes (eigenvectors of $\mathbf{Q}$) converge at different rates. As a result, the local minimum in the gradient direction only matches the global minimum when $\boldsymbol{\delta}$ is an eigenvector of $\mathbf{Q}$. For all other directions, the local minimum over-shoots some modes and under-shoots others. Successive applications of the gradient-based search, with its repeated over-shooting of some modes, causes the descent trajectory to zigzag as it converges, eventually landing close enough to the global minimum to pass the specifications.

The **coordinate descent** is the simplest approach. The descent direction is one of the coordinate axes: $\boldsymbol{\delta}=(\delta_1,\cdots,\delta_M)^T$ where all but one of the elements δ_i ($i=1,\cdots,M$) are zero and the remaining element is either 1 or *j* (for example, $\boldsymbol{\delta}=(1,0,\cdots,0)^T$)**.** The position of the non-zero element is varied to change the direction searched. This is not a good search method when the condition number of $\mathbf{Q}$ is large and is often considerably slower than the gradient descent.

The coordinate search strategy can be used to separate the real and imaginary components. This may be well-suited to a PA operating near saturation because the real and imaginary components correspond roughly to the amplitude and phase correction provided by the DPD. The former is affected more than the latter by the effects of clipping, justifying the use of separate line searches. In the following, references to *M* directions are to be interpreted as 2*M* line searches, where the real and imaginary parts are separate searches.

Conjugate gradient descent uses the gradient for the first direction vector, then derives the remaining *M*-1 direction vectors recursively to be $\mathbf{Q}$-orthogonal, where $\mathbf{Q}$ is the covariance matrix defined earlier. The conjugate gradient method converges to the global minimum after the local minima of the *M* directions are found, when the error surface is quadratic. The $\mathbf{Q}$-orthogonal process is residual-driven and tends to order the most productive directions first, allowing the search to be terminated in fewer than *M* directions if the cost J is low enough. In general, conjugate gradient descent has better convergence properties than coordinate or gradient descents.

An alternative approach is to address the modes individually over *M* iterations. The **eigenvector descent method** uses directions that follow the principal axes of the error surface. This is done by selecting a direction to be the same as an eigenvector of $\mathbf{Q}$: that is, $\boldsymbol{\delta} = \mathbf{v}_i$. Each of the *M* eigenvector directions are searched in sequence starting with the largest eigenvalue $\mathbf{v}_1$. It is similar to a coordinate descent except that the coordinate axes are rotated to align with the principal axes of the elliptical error surface. The convergence is significantly better than the coordinate descent. Both the eigenvalue descent and the conjugate gradient approaches converge in one global cycle when the error surface is quadratic. The conjugate gradient tends to have a lower residual when the global cycle is terminated early.

The error power minimization approach has a disadvantage in that the three measurements of $J(\mathbf{w})$ used to perform the line search are captured over different time intervals. The technique requires that the statistics of the input signal remain constant over the three intervals. The use of a training signal would ensure that this condition is met.

III.3 Feature extraction statistical learning techniques

Principal component analysis (PCA) is a statistical learning technique that is suitable for converting an original set of eventually correlated basis functions (or components) into a new uncorrelated orthogonal basis set called principal components. The principal components are linear combinations of the original basis functions oriented at capturing the maximum variance of the data contained in the data matrix $\boldsymbol{\Phi}$.

Fig. 8-left shows a general example of a PCA transformation considering 2-dimensional data. The original coordinate axes are $\mathbf{u}_1$ and $\mathbf{u}_2$, while the new coordinate axes are $\mathbf{v}_1$ and $\mathbf{v}_2$, corresponding to the two eigenvectors of the original data. The first eigenvector $\mathbf{v}_1$ has a much larger variance $\hat{\sigma}_1$ than the variance $\hat{\sigma}_2$ of the second eigenvector $\mathbf{v}_2$. The variance of the new components corresponds to their associated eigenvalues. Consequently, it is possible to apply dimensionality reduction by discarding the components (eigenvectors) with smaller eigenvalues. In the example of Fig. 8-left, if we keep the principal component $\mathbf{v}_1$ and discard $\mathbf{v}_2$, the information loss will be less harmful than performing the same action in the original coordinate axes, where both $\mathbf{u}_1$ and $\mathbf{u}_2$ present similar variances (i.e., σ_1 and σ_2).

In the case of our DPD system, the principal components of the basis functions (i.e., columns of $\boldsymbol{\Phi}$) are the eigenvectors of $\boldsymbol{\Phi}\boldsymbol{\Phi}^H$. Then, the new transformed matrix (containing the principal components that are orthogonal to each other) is defined as $\hat{\boldsymbol{\Phi}} = \boldsymbol{\Phi}\mathbf{P}_{pca}$, which corresponds to the eigenvectors of the matrix $\boldsymbol{\Phi}\boldsymbol{\Phi}^H$. The transformation matrix $\mathbf{P}_{pca}$ contains the eigenvector of the covariance matrix $\mathbf{Q} = \boldsymbol{\Phi}^H\boldsymbol{\Phi}$. Thanks to the orthogonality property of the resulting transformed matrix, the DPD coefficients' extraction can be carried out with simple dot products (avoiding the Moore–Penrose matrix inversion), since the off-diagonal elements of the transformed matrix $\hat{\mathbf{Q}} = \hat{\boldsymbol{\Phi}}^H\hat{\boldsymbol{\Phi}}$ are zero. In addition, using the adaptive PCA technique [31], it is possible to apply dimensionality reduction in the coefficients estimation by selecting only the minimum necessary number of principal components required to meet the target linearity levels, specified in terms of adjacent channel power ratio (ACPR) and normalized mean square error (NMSE). Fig. 8-right shows the NMSE and ACPR evolution when, at each adaptation iteration, a new orthogonal component is added to the estimation set and, thus, a new coefficient is estimated. Each new added coefficient is estimated independently and reaches a steady state in a few iterations. The estimation is perfectly regularized (i.e., the magnitude of the coefficients is below 1) and after 60 iterations (and

thus a total of 60 components), the targeted NMSE and ACPR are reached and we can stop adding components to the estimation set.

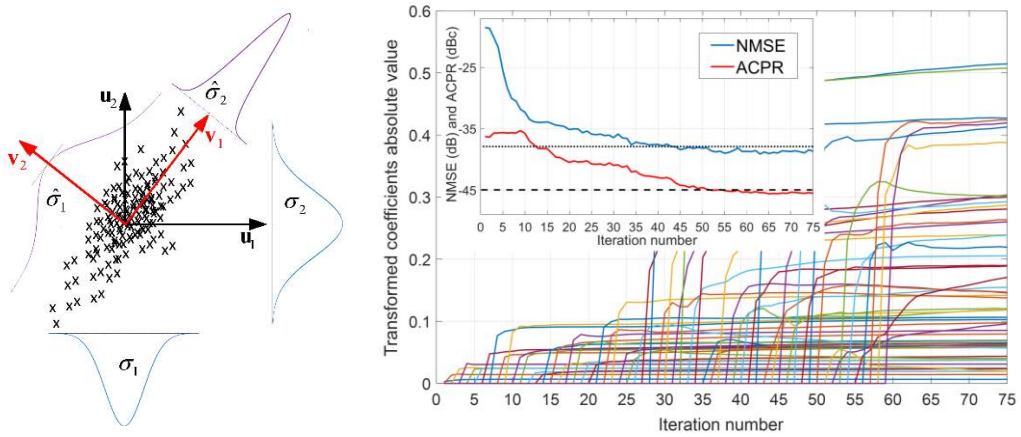

Fig. 8. Example of a PCA transformation (left). Evolution of the magnitude of the DPD coefficients, NMSE, and ACPR considering up to 60 components and 75 iterations (right).

When data becomes very noisy and does not follow a Gaussian distribution, PCA may not be able to provide a robust extraction of the principal components. In this case, **independent component analysis (ICA)** is proposed as a useful alternative method. For instance, ICA is able to identify the independent components of intermixed multi-carrier noisy signals. In [35] the ICA technique is used when distortions are generated due to intermixing of different nonlinear signals. This scenario is similar to intermodulation distortions (IMD) and cross-modulation distortions generated by the PA while driven by multi-carrier and carrier aggregation modulated signals. According to the results provided in [36], thanks to the proposed pruning method based on ICA, the required memory resources for implementing the DPD linearizer are significantly reduced.

Partial least squares (PLS) was introduced by the statistician H. O. A. Wold [32]. Like PCA, PLS is a statistical technique used to construct a new basis of components that are linear combinations of the original basis functions (i.e., $\hat{\mathbf{\Phi}} = \mathbf{\Phi}\mathbf{P}_{pls}$). However, while in PCA the purpose of the transformation (by means of matrix $\mathbf{P}_{pca}$) is to obtain new components that maximize their own variance, in PLS the purpose is to find linear combinations of the original variables that maximize the covariance between the new components and the reference data. This difference enables PLS to outperform PCA in applications such as dimensionality reduction for PA behavioral modeling and DPD linearization.

The procedure for calculating the PLS components is iterative, and it requires that after calculating a new component its information be eliminated from the original basis before calculating the next one. This elimination is called "deflation," and different forms of deflation define several PLS algorithms, one of the most popular of which is SIMPLS [37]. It is worth mentioning that PLS and conjugate gradient methods are very similar. However, while the purpose of using PLS is to create a new transformed basis, the purpose of the conjugate gradient is to perform an iterative search for a set of coefficients that converges to the solution that minimizes a specific quadratic function. PLS and conjugate gradient may have different original goals, but both solutions are obtained by equivalent algorithmic procedures [38]-[39].

The PLS technique is used in [33]-[34] to generate a set of new components from the original basis functions and apply it to a DPD update procedure. By properly selecting the most relevant components from the set, it is possible to guarantee a well-conditioned identification while reducing the number of estimated parameters without loss of accuracy. In addition, thanks to the orthonormality property among the components of the new basis, the matrix inversion operation of the LS Moore–Penrose inverse is significantly simplified.

The **canonical correlation analysis (CCA)** method, introduced by H. Hotelling in 1936, finds linear combinations of variables of a given data set that maximally correlate to the reference data. Like PCA and PLS, CCA is also widely used to reduce the dimension of a given set of data. The main differences among them are discussed in the following:

- PCA finds new orthogonal components with maximal variance among themselves. PCA takes into account the input data only. Its performance is independent of the reference data, which, in the DPD linearization context, is the PA output signal $\mathbf{y}$ or the residual linearization error $\mathbf{e}$.
- PLS finds linear combinations of the original basis functions that maximize the covariance $\boldsymbol{\rho}_{\mathbf{PLS}}$ between the new components $\boldsymbol{\Phi}\mathbf{p}_{\mathbf{i}}$ and the reference data $\mathbf{e}$; this covariance is defined by the expression: $\boldsymbol{\rho}_{\mathbf{PLS}} = \frac{1}{\|\mathbf{e}\|_2}\frac{\langle \boldsymbol{\Phi}\mathbf{p}_{\mathbf{i}}, \mathbf{e}\rangle}{\|\mathbf{p}_{\mathbf{i}}\|_2}$, with $\langle \cdot,\cdot \rangle$ being the inner product, $\|\cdot\|_2$ being the Euclidean norm, and $\mathbf{p}_{\mathbf{i}}$ the transformation vector *i* (from $\mathbf{P}_{pls}$) used to create the specific *i* component of the new basis. This enables PLS to outperform PCA in applications such as dimensionality reduction for PA behavioral modeling and DPD linearization.
- CCA finds new components with maximal correlation coefficient $\boldsymbol{\rho}_{\mathbf{CCA}}$ between the new components and the reference data, being: $\boldsymbol{\rho}_{\mathbf{CCA}} = \frac{1}{\|\mathbf{e}\|_2}\frac{\langle \boldsymbol{\Phi}\mathbf{p}_{\mathbf{i}}, \mathbf{e}\rangle}{\|\boldsymbol{\Phi}\mathbf{p}_{\mathbf{i}}\|_2}$. Like PLS, CCA also depends on both the original and the reference data. Since the correlation coefficient relationship among the components does not depend on the length of the components (unlike in the case of PLS new components), CCA shows better performance than PLS.

Summing up, CCA demonstrates the best reduction capabilities. However, assuming that we handle tall-and-skinny matrices (i.e., many more data samples/equations than basis functions: *N*>>*M*), PLS has the lowest computational cost, $\mathrm{O}(NM)$, while PCA and CCA both have more computational complexity $\mathrm{O}(NM^2)$.

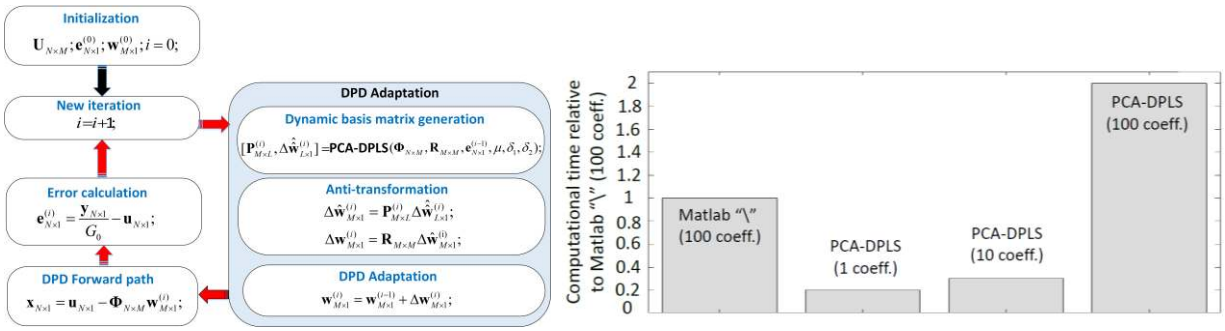

Fig. 9. PCA-DPLS algorithm flow chart (left). Computational time of the PCA-DPLS algorithm considering different number of coefficients and taking as a reference the computational time of MATLAB's backslash operation (right).

A new technique for dynamically estimating and updating the DPD coefficients based on the combination of PCA transformation and dynamic PLS (DPLS) extraction of components was presented in [39]. The proposed PCA-DPLS approach is equivalent to a CCA updating solution, which is optimal in the sense of generating components with maximum correlation. The PCA-DPLS method allows for updating as many coefficients as necessary for achieving the required linearity, and it stops this update when it detects that the DPD basis is not able to further minimize the remaining estimation error. A flow chart of the PCA-PLS algorithm is depicted in Fig. 10-left. This allows reduction of computational costs and of ill-conditioning problems compared to other methods that use a fixed number of coefficients when solving the required LS estimation in the DPD adaptation loop. Fig. 10-right compares the processing time of PCA-DPLS when using 1, 10, and 100 coefficients normalized to the processing time of MATLAB's backslash (or *mldivide* function) operation with 100 coefficients, equivalent to solving LS by means of QR decomposition. For 100 coefficients, MATLAB's backslash operation is roughly 2 times faster than the PCA-DPLS algorithm. However, the PCA-DPLS technique can significantly reduce the number of computed coefficients in the DPD adaptation subsystem while still achieving the same linearity levels as LS. Therefore, by significantly reducing the number of coefficients, the PCA-DPLS processing time is only one-third or less than that of MATLAB's backslash operation. Moreover, in the case of using only 1 coefficient, the PCA-DPLS running time is five times faster than

MATLAB's backslash operation. This is true with high probability since PCA-DPLS is equivalent to CCA when no significant degradation occurs. The predistortion results for each of the methods are shown in Table III [39].

Table III. DPD performance comparison when linearizing a class-J PA with an OFDM-based signal of 80 MHz BW.

DPD updating method	Number of coefficients (max/min)	NMSE [dB]	ACPR [dB]
No DPD	-	-18.6	-36.35
LS	100/100	-40.39	-49.08
CCA	1/1	-40.73	-49.34
PCA-DPLS	10/1	-40.35	-49.33

IV. Conclusion

In this paper we discussed several solutions proposed in the literature for the identification of DPD parameters. First, we addressed some common problems that may lead to an inaccurate coefficient estimation and unstable adaptive DPD linearization. Then, focusing on how to avoid an ill-conditioned estimation and discussing the convergence properties or the possibility of applying dimensionality reduction, we presented an overview of: some stochastic gradient techniques oriented at real-time adaptation; gradient techniques understood as power minimization techniques; and finally, feature extraction techniques, which are statistical approaches used to estimate only a reduced subset of the original parameters, thus reducing the computational complexity as well as benefiting from the associated regularization effects.

Acknowledgements

This work was supported in part by the Spanish Government and FEDER under MICINN projects TEC2017-83343-C4-1-R and TEC2017-83343-C4-2-R and by the Generalitat de Catalunya under Grant 2017 SGR 813.

References

[1] D. Scheurs, M. O'Droma, A. A. Goacher, and M. Gadringer, editors. RF Power Amplifier Behavioural Modeling. Cambridge University Press, 2009.

[2] A. Katz, J. Wood and D. Chokola, "The Evolution of PA Linearization: From Classic Feedforward and Feedback Through Analog and Digital Predistortion," in *IEEE Microwave Magazine*, vol. 17, no. 2, pp. 32-40, Feb. 2016.

[3] P. Roblin, C. Quindroit, N. Naraharisetti, S. Gheitanchi, and M. Fitton, "Concurrent linearization: The state of the art for modeling and linearization of multiband power amplifiers," *IEEE Microw. Mag.*, vol. 14, no. 7, pp. 75–91, Nov. 2013.

[4] C. Fager, T. Eriksson, F. Barradas, K. Hausmair, T. Cunha and J. C. Pedro, "Linearity and Efficiency in 5G Transmitters: New Techniques for Analyzing Efficiency, Linearity, and Linearization in a 5G Active Antenna Transmitter Context," *in IEEE Microwave Magazine*, vol. 20, no. 5, pp. 35-49, May 2019.

[5] 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, April 2015.

[6] P. L. Gilabert, G. Montoro, D. Vegas, N. Ruiz and J. A. Garcia, "Digital Predistorters Go Multidimensional: DPD for Concurrent Multiband Envelope Tracking and Outphasing Power Amplifiers," in *IEEE Microwave Magazine*, vol. 20, no. 5, pp. 50-61, May 2019.

[7] R. N. Braithwaite, "General principles and design overview of digital pre-distortion," in Digital Front-End in Wireless Communications and Broadcasting, F.-L. Luo, Ed. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[8] A N Tikhonov and V Y Arsensin. Solution of ill-posed problems. V H Winston, Washington DC, 1977.

[9] Robert Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society*, Series B, 58:267-288, 1994.

[10] J. Chani-Cahuana, P. N. Landin, C. Fager and T. Eriksson, "Iterative Learning Control for RF Power Amplifier Linearization," in *IEEE Transactions on Microwave Theory and Techniques*, vol. 64, no. 9, pp. 2778-2789, Sept. 2016.

[11] R. N. Braithwaite, "Reducing estimator biases due to equalization errors in adaptive digital predistortion systems for RF power amplifiers," in *2012 IEEE MTT-S Int. Microw. Symp. Dig.*, June 2012, pp. 1–3.

[12] Simon Haykin. *Adaptive Filter Theory, 5th edition*. Pearson, 2014.

[13] G. Montoro, P. L. Gilabert, E. Bertran, A. Cesari and J. A. Garcia, "An LMS-Based Adaptive Predistorter for Cancelling Nonlinear Memory Effects in RF Power Amplifiers," *2007 Asia-Pacific Microwave Conference*, Bangkok, 2007, pp. 1-4.

[14] P. L. Gilabert, G. Montoro and E. Bertran, "FPGA Implementation of a Real-Time NARMA-Based Digital Adaptive Predistorter," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 58, no. 7, pp. 402-406, July 2011.
[15] S. Lee et al., "Digital Predistortion for Power Amplifiers in Hybrid MIMO Systems with Antenna Subarrays," *2015 IEEE 81st Vehicular Technology Conference* (VTC Spring), Glasgow, 2015, pp. 1-5.
[16] P. Gilabert, G. Montoro and E. Bertran, "On the Wiener and Hammerstein models for power amplifier predistortion," *2005 Asia-Pacific Microwave Conference Proceedings*, Suzhou, 2005, pp. 4 pp.-.
[17] Mourad Djamai, Smail Bachir, Claude Duvanaud. Kalman filtering algorithm for on-line memory polynomial predistortion. 38th European Microwave Conference, Oct 2008, Amsterdam, Netherlands. pp. 987 – 990.
[18] D. Falconer and L. Ljung, "Application of Fast Kalman Estimation to Adaptive Equalization," *in IEEE Transactions on Communications*, vol. 26, no. 10, pp. 1439-1446, October 1978, doi: 10.1109/TCOM.1978.1093988.
[19] N. Zheng, Y. Chen, X. Wu and J. Shi, "Digital Predistortion Based on QRD-RLS Algorithm and Its Implementation Using FPGA," *2009 First International Conference on Information Science and Engineering*, Nanjing, 2009, pp. 200-203.
[20] S. D. Muruganathan and A. B. Sesay, "A QRD-RLS-Based Predistortion Scheme for High-Power Amplifier Linearization," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 53, no. 10, pp. 1108-1112, Oct. 2006.
[21] Y. Li and X. Zhang, "Adaptive digital predistortion based on MC-FQRDRLS algorithm using indirect learning architecture," *in Proc. 2nd Int. Conf. Adv. Comput. Control*, vol. 4. Mar. 2010, pp. 240–242.
[22] R. N. Braithwaite, "Fixed Point Considerations for Digital Predistortion of a RF Power Amplifier Using Recursive Least Square (RLS) Estimation," *2019 IEEE Topical Conference on RF/Microwave Power Amplifiers for Radio and Wireless* Applications (PAWR), Orlando, FL, USA, 2019, pp. 1-3.
[23] David G. Luenberger, "Linear and nonlinear programming," Reading MA: Addison Wesley, 1984.
[24] N. Kelly and A. Zhu, "Low-Complexity Stochastic Optimization-Based Model Extraction for Digital Predistortion of RF Power Amplifiers," in *IEEE Transactions on Microwave Theory and Techniques*, vol. 64, no. 5, pp. 1373-1382, May 2016, doi: 10.1109/TMTT.2016.2547383.
[25] N. Kelly and A. Zhu, "Direct Error-Searching SPSA-Based Model Extraction for Digital Predistortion of RF Power Amplifiers," in *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, no. 3, pp. 1512-1523, March 2018.
[26] J. Spall, "An overview of the simultaneous perturbation method for efficient optimization," *John Hopkins APL Tech. Dig.*, vol. 19, no. 4, pp. 482–492, 1998.
[27] R. N. Braithwaite, "Descent-based coefficient estimator for analog predistortion of a dual-band RF transmitter," *2017 IEEE MTT-S International Microwave Symposium* (IMS), Honololu, HI, 2017, pp. 1995-1998.
[28] H. Jiang, X. Yu, and P. A. Wilford, "Digital predistortion using stochastic conjugate gradient method," *IEEE Transactions on Broadcasting*, vol. 58, No. 1, March 2012.
[29] P. L. Gilabert et al., "Order reduction of wideband digital predistorters using principal component analysis," *2013 IEEE MTT-S International Microwave Symposium Digest (MTT)*, Seattle, WA, 2013, pp. 1-7.
[30] 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," *2018 IEEE/MTT-S International Microwave Symposium - IMS*, Philadelphia, PA, 2018, pp. 160-163.
[31] D. López–Bueno, Q. A. Pham, G. Montoro and P. L. Gilabert, "Independent Digital Predistortion Parameters Estimation Using Adaptive Principal Component Analysis," in *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, no. 12, pp. 5771-5779, Dec. 2018.
[32] H. Wold, "Partial least squares," Encyclopedia of Statistical Sciences, vol. 6, pp. 581-591, 1985.
[33] Q. A. Pham, D. López-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," in *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, no. 12, pp. 5143-5150, Dec. 2018.
[34] Q. A. Pham, D. López-Bueno, G. Montoro and P. L. Gilabert, "Dynamic Selection and Update of Digital Predistorter Coefficients for Power Amplifier Linearization," *2019 IEEE Topical Conference on RF/Microwave Power Amplifiers for Radio and Wireless Applications (PAWR)*, Orlando, FL, USA, 2019, pp. 1-4.
[35] P. Jaraut, G. C. Tripathi, M. Rawat and P. Roblin, "Independent component analysis for multi-carrier transmission for 4G/5G power amplifiers," *2017 89th ARFTG Microwave Measurement Conference (ARFTG)*, Honololu, HI, 2017, pp. 1-4.
[36] P. Jaraut, M. Rawat and P. Roblin, "Digital predistortion technique for low resource consumption using carrier aggregated 4G/5G signals," in *IET Microwaves, Antennas & Propagation*, vol. 13, no. 2, pp. 197-207, 6 2 2019.
[37] S. de Jong, "SIMPLS: An alternative approach to partial least squares regression," Chemometrics and Intelligent Laboratory Systems, vol. 18, no. 3, pp. 251–263, Mar. 1993.
[38] 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.
[39] Q. A. Pham, G. Montoro, D. López-Bueno and P. L. Gilabert, "Dynamic Selection and Estimation of the Digital Predistorter Parameters for Power Amplifier Linearization," in *IEEE Transactions on Microwave Theory and Techniques*, vol. 67, no. 10, pp. 3996-4004, Oct. 2019.