Complexity-Reduced Model Adaptation for Digital Predistortion of RF Power Amplifiers With Pretraining-Based Feature Extraction

Yue Li, *Member, IEEE,* Xiaoyu Wang, *Graduate Student Member, IEEE,* and Anding Zhu, *Senior Member, IEEE*

***Abstract*—In this article, we present a new method to reduce the model adaptation complexity for digital predistortion (DPD) of radio frequency (RF) power amplifiers (PAs) under varying operating conditions, using pretrained transformation of model coefficients. Experimental studies show that the PA behavior variations can be effectively tracked using a small number of "transformed" coefficients, even with large deviations in its output characteristics. Based on this discovery, to avoid re-extracting all the original coefficients every time when the operating condition changes, we propose to conduct a one-time offline pretraining stage to extract the common features of PA behaviors under different operating conditions first. The online model adaptation process will then only need to identify a small number of transformed coefficients, which can result in a drastic reduction in the computational complexity of model adaptation process. The proposed solution is validated by experimental results considering varying signal bandwidth and output power levels on a high-efficiency Gallium Nitride Doherty PA, where the computational complexity is significantly reduced and the system performance is not compromised.**

***Index Terms*—Digital predistortion, model adaptation, power amplifier, principal component analysis, transfer learning, Volterra series**

I. Introduction

With the development of modern wireless technologies, digital predistortion (DPD) is widely deployed to mitigate the nonlinear distortions induced by the transmitters, especially the radio frequency (RF) power amplifiers (PAs) [1], [2]. Today, driven by the endless pursuit of higher spectral and energy efficiency, the evolution towards the next generation cellular communication technologies, 5G, is demanding a more heterogeneous and more efficient network to fulfill the speed and coverage requirements [3]. Thus, to comply with the different linearity and efficiency specifications, DPD is expected to deliver better linearization performance with lower complexity in diverse application scenarios.

A modern DPD system [4] usually consists of three key parts: the predistorter, the data acquisition receiver and the model extraction/adaptation unit. The predistorter is implemented in the transmitter path with a DPD model that produces the predistorted signal in real time. The data acquisition receiver captures the signals from the PA output, which are used by the model adaptation unit to extract the model coefficients. Since model adaptation is only activated when the PA characteristics change and the DPD coefficients need to be updated, the power consumption of model extraction has not been a big concern.

When DPDs are deployed to linearize 5G transmitters, several issues arise. Firstly, the introduction of small-cell base stations requires DPD with low hardware complexity and low power consumption, which makes high-complexity model adaptation algorithms unfavorable [5]. Secondly, new frame structures in 5G allow more flexible resource allocation [3], leading to rapid system reconfiguration and fast varying PA behavior. To cope with this challenge, faster adaptation is essential. Thirdly, the use of beamforming techniques further complicates the existing problems, because we need to deal with more complex nonlinearities and rapidly changing beam directions [6]. Therefore, model adaptation is expected to be an important concern in future system design.

To date, the majority of DPD models are linear in parameters and thus linear system identification methods, e.g., the least squares (LS) algorithm, can be deployed for extracting the model coefficients [7]. Despite the theoretical simplicity, the high algorithmic complexity of LS makes it an expensive solution for DPD model adaptation. While several techniques [8]–[13] have been proposed to reduce the complexity of LS, it still grows rapidly with the number of coefficients. Since future DPD models will inevitably involve a large number of coefficients to address the complex PA nonlinearities [14], the complexity of the LS-based methods is expected to increase further. Some alternatives to the LS solution, e.g., least mean squares (LMS) [15], [16] and stochastic optimization [17], [18], can achieve lower complexity per iteration, but they suffer from slow convergence speed and potential stability issues. Therefore, in applications targeting low hardware complexity, like small-cell base stations, applying existing model adaptation algorithms faces many challenges.

To alleviate the complexity issue, some adaptation strategies have been proposed to reduce the updating frequency of DPD. One method is to record the previously extracted coefficients. When the same or similar operating condition occurs again, they can be retrieved and copied to DPD [19] or used as initial states for further adaptations [20]. Another method is

Manuscript received Apr 15, 2020; revised Jul 10, 2020 and Sep 28, 2020; accepted Oct 24, 2020. This work was supported by the Science Foundation Ireland under Grant Numbers 13/RC/2077, 17/NSFC/4850 and 16/IA/4449. (*Corresponding author: Yue Li*)

The authors are with the School of Electrical and Electronic Engineering, University College Dublin, Dublin, D04 V1W8, Ireland. (e-mail: yue.li@ucd.ie; xiaoyu.wang1@ucdconnect.ie; anding.zhu@ucd.ie)

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

Digital Object Identifier XXXXXXXXXXXXX

to explicitly express the DPD coefficients as a function of physical parameters [21]–[23]. Thus, once we observe changes on the parameters, the DPD coefficients can be updated without new feedback data. Yet, these methods all require knowing the current PA state, so the estimation accuracy may degrade if such information cannot be acquired accurately or if the PA behavior is changed significantly by other unobserved effects.

After carefully investigating the variations of PA behavior under different operating conditions, we find that these variations are usually highly correlated and they can be accurately modeled with a small number of common features. Based on this discovery, in this work, we propose a new method to reduce the complexity of DPD model adaptation based on the concept of "transfer learning" [24] that utilizes pretrained transformation of model coefficients. The method is split into two phases: the first phase is called "pretraining" that is conducted offline and used to determine a compact basis function set based on singular value decomposition (SVD) of the ideal output of the DPD; the second phase is the normal model extraction that can be conducted with online adaptation of the transformed coefficients of the model. These transformed coefficients are then transformed back to the original coefficients of the full model basis and are applied to the DPD model. By pretraining the DPD model under various representative operating conditions, we can effectively extract common features to model PA characteristics under varying operating conditions with a small number of coefficients. In real time operation, we therefore only need to update the transformed coefficients instead of the original model coefficients, which significantly reduces the number of coefficients to be estimated in DPD model adaptation. That can lead to a significant reduction in the computational complexity in the model adaptation process. The dramatic decrease of the number of coefficients to be estimated also allows significant reduction of the number of training samples needed to identify the coefficients, which can further reduce the computational complexity of model adaptation.

The rest of the paper is organized as follows: Section II presents a review of DPD model adaptation and feature extraction techniques, as well as an experimental analysis of PA characteristics under varying operating conditions. Section III develops the novel pretraining framework. We discuss both offline feature extraction algorithms and online model adaptation methods in detail. In Section IV, we present the experimental results and complexity analysis. Finally, a conclusion is drawn in Section V.

II. DPD Model Adaptation

DPD compensates for PA nonlinear effects by applying an inverted model of the PA to the input signal at digital baseband before amplification. In the past years, a wide range of DPD models have been developed. If the model is linear in its parameters, it can be expressed in matrix format as below,

$$\mathbf{u} = \mathbf{X}\mathbf{c}, \tag{1}$$

where $\mathbf{u}$ is the predistorted signal vector consisting of N data samples,

$$\mathbf{u} = [\tilde{u}(N), \tilde{u}(N-1), \cdots,]^T, \tag{2}$$

$\mathbf{c}$ is a vector including all Q model coefficients,

$$\mathbf{c} = [c(1), c(2), \cdots, c(Q)]^T, \tag{3}$$

and $\mathbf{X}$ is an N-by-Q regression matrix containing all basis functions constructed with the input signal samples $\tilde{x}(n)$. For example, if the memory polynomials (MP) model [25] is used, the basis functions have the form of $|\tilde{x}(n-m)|^{k-1}\tilde{x}(n-m)$, where k is the nonlinearity order and m is the delay number, and $\mathbf{X}$ can be constructed as

$$\mathbf{X} = \begin{bmatrix} \tilde{x}(N) & \tilde{x}(N-1) & \cdots & |\tilde{x}(N)|\tilde{x}(N) & \cdots \\ \tilde{x}(N-1) & \tilde{x}(N-2) & \cdots & |\tilde{x}(N-1)|\tilde{x}(N-1) & \cdots \\ \vdots & \vdots & \ddots & \vdots & \ddots \end{bmatrix} \tag{4}$$

A. Conventional Model Adaptation

To obtain the values of the DPD coefficients, a small fraction of the transmit signal is usually transferred back to baseband via a feedback loop and a model extraction/adaptation algorithm can be applied. For instance, if the LS is used, the value of the model coefficients can be found via

$$\hat{\mathbf{c}} = \left(\mathbf{X}^H\mathbf{X}\right)^{-1}\mathbf{X}^H\hat{\mathbf{u}}, \tag{5}$$

where $\hat{\mathbf{u}}$ is the target model output and $\hat{\mathbf{c}}$ is the estimated model coefficients.

In practical wireless systems, the PA needs to operate at different conditions to cope with the varying environment and meet different transmission requirements. The characteristics of the PA can change with different factors, such as signal bandwidth, modulation schemes and power levels. To maintain the linearity, the DPD model must be frequently updated to track these variations. Updating the model coefficients usually requires matrix operations, as shown in (5). The computational complexity of these operations depends on the number of DPD coefficients and the number of data samples used. If a large number of coefficients or data samples are involved, the computational complexity can be very high.

B. Model Adaptation Based on Feature Extraction Under Fixed Condition

To reduce computational complexity, one solution is to conduct model order reduction by using a feature extraction scheme. The aim of feature extraction is to compress the information of the original basis functions into a new feature matrix. For example, in principal component analysis (PCA)-based methods [10], [11], it is achieved by using the SVD of $\mathbf{X}$, or equivalently the eigendecomposition of $\mathbf{X}^H\mathbf{X}$,

$$\mathbf{X} = \mathbf{U}_\mathbf{X}\mathbf{\Sigma}_\mathbf{X}{\mathbf{V}_\mathbf{X}}^H, \tag{6}$$

where $\mathbf{U}_\mathbf{X}$ and $\mathbf{V}_\mathbf{X}$ are orthonormal matrices and $\mathbf{\Sigma}_\mathbf{X}$ is a diagonal matrix. $\mathbf{U}_\mathbf{X}$ includes the principal components of $\mathbf{X}$, whose importance is measured by the corresponding diagonal

elements of $\mathbf{\Sigma_X}$. By keeping the P most important principal components and discarding the rest, we obtain a new N-by-P feature matrix $\mathbf{U_X^{(P)}}$ which is the first P columns of $\mathbf{U_X}$. By using the feature matrix $\mathbf{U_X^{(P)}}$, instead of $\mathbf{X}$, to calculate the LS solution, the computational complexity can be reduced, because of its smaller dimension N-by-P, as compared to N-by-Q of $\mathbf{X}$.

Nevertheless, the existing solutions [10]–[12] mainly focus on deriving the feature transformation under one specific operating condition, so the effectiveness of the extracted features may be limited when the operating condition changes. For example, the PCA transformation matrix [10], [11] can only be reused for signals with similar properties, and the partial least squares (PLS) transformation in [12] needs to be recalculated whenever the optimization target changes. To take into account the various operating conditions, it is desirable to seek better feature extraction methods.

C. Feature Extraction Under Varying Conditions

From the previous studies [21]–[23], [26], we can see that, though there are differences, the nonlinear behaviors of the PA under different operating conditions are strongly correlated. For instance, in [21], the static characteristics of the PA and the dynamic changes under different power levels are modeled using two sets of coefficients sharing the same basis functions.

To systematically evaluate the correlation, we present an experimental study by varying the PA output power as an example. To help readers reproduce the results, we use a publicly available platform, RF WebLab [27], [28], to conduct the test. A standard Gallium Nitride (GaN) PA was used and the output power of the PA was changed from 25 dBm to 33 dBm with a 1 dB step size. For comparison, we also set another smaller sweeping range, 31 dBm to 33 dBm with a 0.25 dB step size. The two sweeping ranges were set to have the same number of test points. The test signal was a 20 MHz Long-Term Evolution (LTE) signal with 6.5 dB peak-to-average power ratio (PAPR). Two representative AM-AM and AM-PM curves from the test are shown in Fig. 1, where we can see that the PA behaves differently at different power levels with the same input signal. It is worth mentioning that the simple results shown in Fig. 1 are for illustration only and they facilitate interested readers to reproduce the results since RF WebLab is publicly available. More complex tests are given in Section IV.

To extract the correlated components between the different PA output signals, we employ the PCA technique. Different from [10], [11] where PCA is applied on the basis function matrix $\mathbf{X}$, here the captured signals are collected into a matrix $\mathbf{Y_0}$ where each column represents the PA output captured at one power level and the SVD-based PCA is performed on $\mathbf{Y_0}$,

$$\mathbf{Y_0} = \mathbf{U_Y}\mathbf{\Sigma_Y}\mathbf{V_Y}^H, \tag{7}$$

where $\mathbf{U_Y}$ and $\mathbf{V_Y}$ are orthonormal matrices and $\mathbf{\Sigma_Y}$ is a diagonal matrix. $\mathbf{U_Y}$ contains the principal components of $\mathbf{Y_0}$, and the diagonal entries of $\mathbf{\Sigma_Y}$, Σ_i's, measure the magnitude of each principal component. As the analysis is conducted on the PA output data, it exhibits fundamental differences

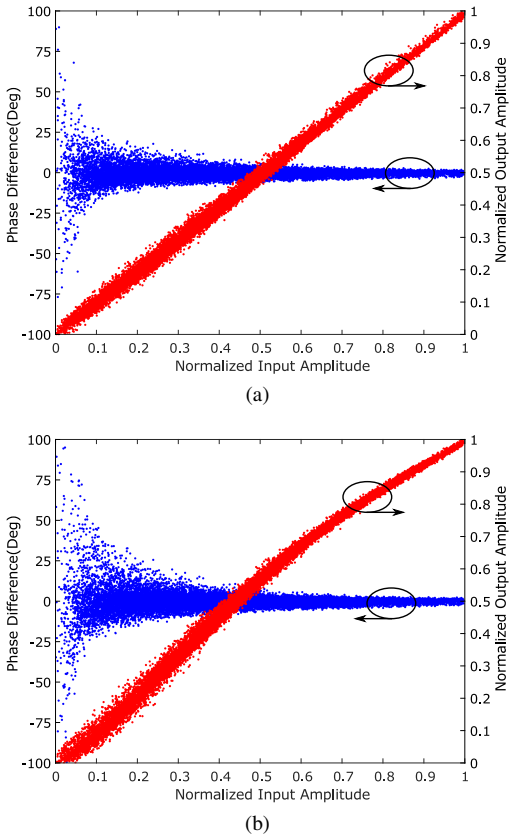


Fig. 1. AM-AM and AM-PM characteristics at (a) 25 dBm and (b) 33 dBm.

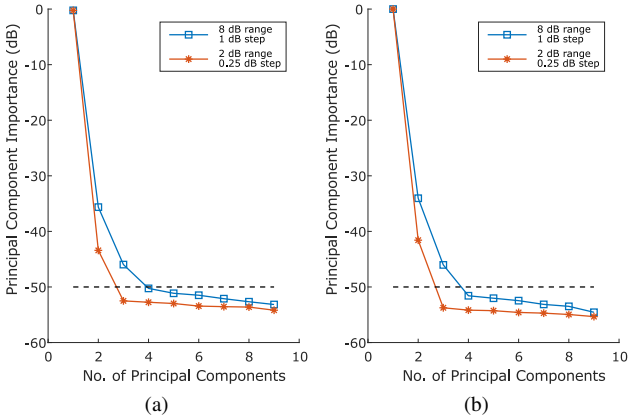


Fig. 2. Importance of principal components in (a) PA output without DPD and (b) "ideal" DPD output obtained by ILC.

with the prior arts. In [10], [11], Σ_i is used to evaluate the relative importance of each basis function, but in this work, Σ_i indicates the significance of different dynamic effects caused by the changing operating conditions. To better illustrate the results, Σ_i's are normalized by the total sum of squares to show the relative importance,

$$\text{Imp}_i = \frac{|\Sigma_i|^2}{\|\mathbf{\Sigma_Y}\|^2}, \tag{8}$$

where $\|\cdot\|$ represents the Frobenius norm.

The relative importance of all principal components of both sweeps are depicted in decibel in Fig. 2(a). It can be

clearly seen that, most components are unimportant and thus contribute little to describing the changes in $\mathbf{Y_0}$. If we set -50 dB as the threshold and discard all components below it, we will end up with only 2 components for the 2 dB sweep, and 3 components for the 8 dB sweep.

Using the same methodology, we further examine if the DPD response, the inverse of PA behavior, has the similar property. To obtain the "ideal" DPD output, the iterative learning control (ILC) technique [29] was employed in every operating condition. After performing PCA on the predistorted signal, the results are shown in Fig. 2(b). It can be easily seen that the distribution of principal component importance is almost identical to that in Fig. 2(a).

From the tests, we can conclude that, for a given PA, there exists a small set of principal components whose linear combination can accurately model the PA output or ideal predistorted signals at varying conditions. Therefore, for both PA and the inverse characteristics, the variation caused by changes of operating conditions can be characterized by just a few principal components, despite the numerous possible conditions in the sweep range.

III. Pretraining-Based DPD

The previous findings suggest that the desired DPD output signals under different operating conditions are composed of only a small number of common components. Nevertheless, this property is difficult to exploit in existing model adaptation methods, due to their inherent inability to cope with varying operating conditions. In this work, we develop a pretraining-based DPD scheme to take full advantage of the similarity between DPD responses under different operating conditions and significantly reduce the computational complexity in DPD model adaptation.

A. The Proposed Framework

To incorporate the different dynamic effects, we propose to employ a feature extraction approach and use the different features to represent the various dynamics. The DPD function can be expressed as

$$\mathbf{u} = \mathbf{Xc} = \mathbf{XAw}, \tag{9}$$

where $\mathbf{A}$ is a feature transformation matrix and $\mathbf{w}$ represents the transformed coefficients. The original DPD model can thus be transformed into a new structure, as depicted in Fig. 3. In this structure, if we let the same transformation matrix $\mathbf{A}$ be shared by all operating conditions and allow individual condition to have its own $\mathbf{w}$, we only need to update $\mathbf{w}$ when the system operating condition changes.

The basic procedures to realize this idea is shown in Fig. 4, which consist of two stages. In this framework, the pretraining stage is performed only once in an offline setting and the adaptation stage is implemented in the actual DPD system to update the DPD model. The system architecture is depicted in Fig. 5, which highlights the different setups of the two stages.

In the pretraining stage, we capture the target data, i.e., the ideal DPD output signals, by sweeping over different operating conditions. In each case, the PA circuit and the training signals

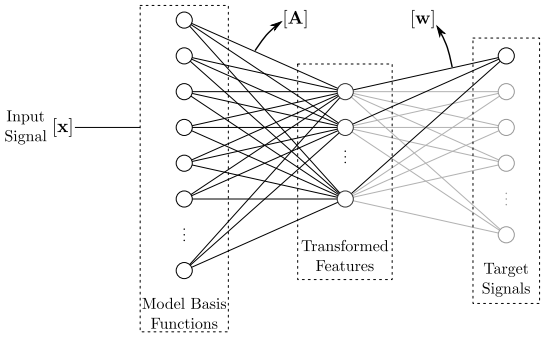


Fig. 3. New model structure under varying operating conditions.

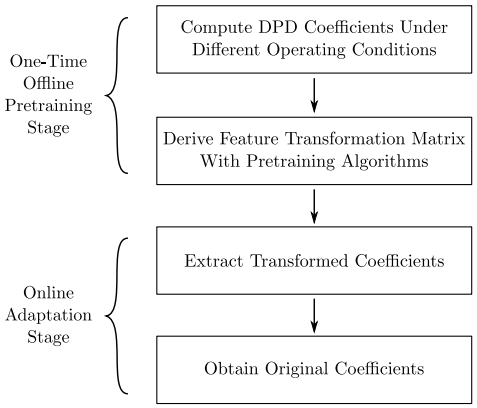


Fig. 4. Procedures of the proposed pretraining-based DPD model adaptation.

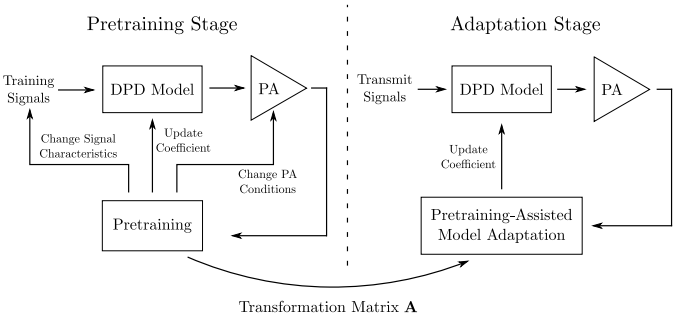


Fig. 5. Pretraining-assisted model adaptation architecture.

can change to set up a proper operating condition, and the DPD model may be updated iteratively to find the optimum predistorted signal for this specific condition. To match the situation in practical applications, the sweeping is expected to include all factors of interest. With the sweeping data, we implement offline training algorithms, as will be detailed shortly, to extract the important features. If most representative cases are covered and the learned feature transformation can perform well in most cases, the model can also be generalized to cover untested cases.

With the assistance of pretraining, the original regression matrix $\mathbf{X}$ can be transformed to $\mathbf{Z}$,

$$\mathbf{Z} = \mathbf{XA}. \tag{10}$$

The DPD model can then be expressed as

$$\mathbf{u} = \mathbf{Zw}. \tag{11}$$

In the model adaptation stage, we only need to train the transformed coefficients vector $\mathbf{w}$, which is expected to have a much smaller size than that of the original model coefficients vector $\mathbf{c}$. After extracting the new coefficients $\mathbf{w}$, we can conveniently transform them back to $\mathbf{c}$ and apply to the predistorter.

B. The Pretraining Algorithms

In the proposed architecture, the key issue is to optimize the feature transformation matrix $\mathbf{A}$ which should minimize the difference between the model output and the desired output under all pretraining cases. Additionally, it is also helpful to develop a concrete guideline to choose the value of P, the number of transformed features. In this part, we propose novel pretraining algorithms to fulfill the aforementioned two goals.

1) Pretraining Using Fixed Training Signals: If the same input signals are employed during pretraining, the optimization of $\mathbf{A}$ can be translated into solving

$$\min_{\mathbf{A}} \|\mathbf{XAW} - \mathbf{Y}\|^2, \tag{12}$$

where $\mathbf{Y}$ contains the target data under all pretraining conditions, and $\mathbf{W}$ includes the new coefficients under all cases. Perform QR decomposition on $\mathbf{X}$,

$$\mathbf{X} = \mathbf{QR}, \tag{13}$$

where $\mathbf{Q}$ is an orthonormal matrix and $\mathbf{R}$ is an upper triangular matrix. Then we can express $\mathbf{Y}$ as

$$\mathbf{Y} = \mathbf{QD} + \mathbf{E} \tag{14}$$

where $\mathbf{D}$ collects the transformed coefficients under all conditions and $\mathbf{E}$ is the modeling residue orthogonal to $\mathbf{Q}$. Thus, the minimization objective can be rewritten as

$$\min_{\mathbf{A}} \|\mathbf{QRAW} - \mathbf{QD} - \mathbf{E}\|^2. \tag{15}$$

Because of the orthogonality, it can be further simplified to

$$\min_{\mathbf{A}} \|\mathbf{RAW} - \mathbf{D}\|^2. \tag{16}$$

Thus, it can be solved by performing SVD on matrix $\mathbf{D}$

$$\mathbf{D} = \mathbf{U_D}\mathbf{\Sigma_D}{\mathbf{V_D}}^H, \tag{17}$$

and make the following assignment

$$\mathbf{RA}' = \mathbf{U_D} \tag{18}$$

$$\mathbf{W}' = \mathbf{\Sigma_D}{\mathbf{V_D}}^H. \tag{19}$$

The importance of principal features can be measured by the diagonal elements of $\mathbf{\Sigma_D}$ and P can be chosen by setting an error threshold for them. One reasonable choice is to set the threshold 10 dB below the desired NMSE value. It is also possible to include a few more features for potentially better generalization performance in untested cases.

Finally, by keeping the P most significant principal components, we have

$$\mathbf{A} = \mathbf{R}^{-1}\mathbf{U}_{\mathbf{D}}^{(\mathbf{P})}, \tag{20}$$

where $\mathbf{U}_{\mathbf{D}}^{(\mathbf{P})}$ is the first P columns of $\mathbf{U_D}$.

The procedures are summarized in Algorithm 1.

Algorithm 1 QR-SVD Pretraining Algorithm

Input: $\mathbf{X}$, $\mathbf{Y}$

Output: $\mathbf{A}$

1: Perform QR decomposition on $\mathbf{X}$: $\mathbf{X} = \mathbf{QR}$
2: $\mathbf{D} = \mathbf{Q^H Y}$
3: Perform SVD on $\mathbf{D}$: $\mathbf{D} = \mathbf{U_D}\mathbf{\Sigma_D}{\mathbf{V_D}}^H$
4: Determine the number of required principal components P by checking the diagonal elements of $\mathbf{\Sigma_D}$
5: Form $\mathbf{U}_{\mathbf{D}}^{(\mathbf{P})}$ using the first P columns of $\mathbf{U_D}$
6: $\mathbf{A} = \mathbf{R}^{-1}\mathbf{U}_{\mathbf{D}}^{(\mathbf{P})}$

2) Pretraining Using Varying Training Signals: In real system operations, the input signal characteristics, e.g., bandwidth or PAPR, may change with the system configurations. Thus, it is necessary to train $\mathbf{A}$ using input signals of different characteristics. To accommodate different input signals, we generalize the objective function to

$$\min_{\mathbf{A}} \sum_t \|\mathbf{X_t AW_t} - \mathbf{Y_t}\|^2, \tag{21}$$

where the subscript t is the index of different input signals. Note that each norm has separate regression matrices, model coefficients and target data, but the same feature transformation matrix.

Following a similar derivation in (12) to (16), we have

$$\min_{\mathbf{A}} \sum_t \|\mathbf{R_t AW_t} - \mathbf{D_t}\|^2. \tag{22}$$

Based on the property of QR decomposition, we have

$$\mathbf{D_t} = \mathbf{R_t C_t}. \tag{23}$$

So equivalently we optimize

$$\min_{\mathbf{A}} \sum_t \|\mathbf{R_t AW_t} - \mathbf{R_t C_t}\|^2, \tag{24}$$

where $\mathbf{C_t}$ is the coefficients of the original model before feature extraction. Due to the complex structure, it cannot be directly optimized using SVD. To simplify the optimization, we approximately treat all $\mathbf{R_t}$'s as equal. Though the $\mathbf{R_t}$'s are in general different because of the different signal characteristics, using an approximated $\mathbf{R}$ has little influence on performance [30]. Thus, we assume

$$\mathbf{R_t} \approx \bar{\mathbf{R}}, \tag{25}$$

where $\bar{\mathbf{R}}$ can be computed by performing QR decomposition on the combined regression matrix $\bar{\mathbf{X}}$,

$$\bar{\mathbf{X}} = \bar{\mathbf{Q}}\bar{\mathbf{R}}, \tag{26}$$

where

$$\bar{\mathbf{X}} = \left[{\mathbf{X_1}}^T, {\mathbf{X_2}}^T, \cdots\right]^T. \tag{27}$$

Replacing $\mathbf{R_t}$ with $\bar{\mathbf{R}}$, the objective function can be rearranged as

$$\min_{\mathbf{A}} \sum_t \|\bar{\mathbf{R}}\mathbf{AW_t} - \bar{\mathbf{R}}\mathbf{C_t}\|^2 = \min_{\mathbf{A}} \|\bar{\mathbf{R}}\mathbf{AW} - \bar{\mathbf{R}}\mathbf{C}\|^2, \tag{28}$$

Algorithm 2 Generalized QR-SVD Pretraining Algorithm

Input: all $\mathbf{X_t}$'s, all $\mathbf{Y_t}$'s
Output: $\mathbf{A}$

$\bar{\mathbf{X}} = \left[\mathbf{X_1}^T, \mathbf{X_2}^T, \cdots\right]^T$
Perform QR decomposition on $\bar{\mathbf{X}}$: $\bar{\mathbf{X}} = \bar{\mathbf{Q}}\bar{\mathbf{R}}$
for all t **do**
  $\mathbf{C_t} = \left(\mathbf{X_t}^H\mathbf{X_t}\right)^{-1}\mathbf{X_t}^H\mathbf{Y_t}$
end for
$\mathbf{C} = [\mathbf{C_1}, \mathbf{C_2}, \cdots]$
$\mathbf{D} = \bar{\mathbf{R}}\mathbf{C}$
Perform SVD on $\mathbf{D}$: $\mathbf{D} = \mathbf{U_D}\mathbf{\Sigma_D}\mathbf{V_D}^H$
Determine the number of required principal components P by checking the diagonal elements of $\mathbf{\Sigma_D}$
Form $\mathbf{U_D^{(P)}}$ using the first P columns of $\mathbf{U_D}$
$\mathbf{A} = \bar{\mathbf{R}}^{-1}\mathbf{U_D^{(P)}}$

where $\mathbf{W}$ and $\mathbf{C}$ are the concatenation of all $\mathbf{W_t}$'s and $\mathbf{C_t}$'s. By setting

$$\mathbf{D} = \bar{\mathbf{R}}\mathbf{C}, \tag{29}$$

SVD, as in (17), can then be used to calculate $\mathbf{U_D}$ and $\mathbf{\Sigma_D}$. Finally, we can choose a proper value of P by examining $\mathbf{\Sigma_D}$ and solve $\mathbf{A}$ by

$$\mathbf{A} = \bar{\mathbf{R}}^{-1}\mathbf{U_D^{(P)}}. \tag{30}$$

The full procedures are summarized in Algorithm 2.

C. Adaptation of Transformed Features

After developing procedures to extract effective features for DPD, model adaptation can be greatly simplified. The pretrained DPD model is expressed as that in (11), so we can extract the coefficients simply by LS,

$$\hat{\mathbf{w}} = \left(\mathbf{Z}^H\mathbf{Z}\right)^{-1}\mathbf{Z}^H\mathbf{u}, \tag{31}$$

and obtain the coefficients of the original model by

$$\hat{\mathbf{c}} = \mathbf{A}\hat{\mathbf{w}}. \tag{32}$$

As pretraining directly simplifies the LS method, it can be applied to both indirect learning and closed-loop estimation methods.

Due to the reduction in the number of coefficients, the computational complexity can be reduced. Moreover, in linear system identification algorithms, the required number of data samples is proportional to the number of parameters to be estimated [31]. Thus, with pretraining-assisted adaptation, we can use fewer samples for LS extraction without affecting estimation accuracy, which simultaneously reduces the computational complexity of adaptation algorithm and the power consumption of the analog/mixed-signal components in the DPD data acquisition receiver.

In addition, by examining the derivation of $\mathbf{Z}$, we have

$$\mathbf{Z} = \mathbf{X}\mathbf{A} = \mathbf{Q}\mathbf{R}\bar{\mathbf{R}}^{-1}\mathbf{U_D^{(P)}} \approx \mathbf{Q}\mathbf{U_D^{(P)}}. \tag{33}$$

As both $\mathbf{Q}$ and $\mathbf{U_D^{(P)}}$ are near orthonormal, $\mathbf{Z}$ is approximately orthonormal as well. Thus, the transformation can also improve the conditioning of the model identification.

We also wish to point out that the proposed scheme is independent of existing model adaptation methods [8], [9], [11]–[13] in that the reduced number of updated coefficients is achieved by extracting prior knowledge of the PA under test. As will be shown in Section III-D, it is also possible to combine the pretraining framework with these techniques to achieve further complexity reduction.

D. Co-design With Other Complexity Reduction Methods

If the closed-loop adaptation method is employed, we do not need to accurately calculate the matrix inversion in the LS solution because of its iterative nature [32]. Thus, the computational complexity can be further reduced by combining pretraining with other existing methods, e.g., [9], [12].

As a brief illustration, we take the CM-iDLA method [9] as an example. In [9], the covariance matrix in LS is precalculated using a large amount of data in advance. To apply this method, we can collect a large number of input samples and calculate

$$\mathbf{H} = \left(\mathbf{Z'}^H\mathbf{Z'}\right)^{-1}, \tag{34}$$

where $\mathbf{Z}'$ is obtained in the same way as $\mathbf{Z}$ but using a large training data size L_0.

The target for LS is the difference between the actual PA output and the desired output, $\mathbf{e} = \mathbf{y} - \mathbf{x}$, so we have

$$\mathbf{w^{(m+1)}} = \mathbf{w^{(m)}} - \lambda\Delta\mathbf{w}_{\text{LS}}^{\mathbf{(m)}}, \tag{35}$$

where

$$\Delta\mathbf{w}_{\text{LS}}^{\mathbf{(m)}} = \frac{L_0}{L}\mathbf{H}\mathbf{Z}^H\mathbf{e} = \frac{L_0}{L}\mathbf{H}\mathbf{A}^H\mathbf{X}^H\mathbf{e}, \tag{36}$$

L is the size of training data and λ is the learning rate. The original model coefficients can be calculated by (32).

It is worth mentioning that though the final update equation is similar to the original version in [9], by employing the pretraining method, the amount of training data can be reduced significantly, which can be directly translated to the reduction of computational complexity.

E. Co-design With LMS Estimators

The principle of pretraining can also be applied to LMS-type adaptation algorithms. The update equation for conventional LMS algorithm is

$$\mathbf{c^{(m+1)}} = \mathbf{c^{(m)}} - \mu\Delta\mathbf{c}_{\text{LMS}}^{\mathbf{(m)}}, \tag{37}$$

where

$$\Delta\mathbf{c}_{\text{LMS}}^{\mathbf{(m)}} = \mathbf{X}(m)^H e(m), \tag{38}$$

$\mathbf{X}(m)$ includes all basis functions for the mth sample, $e(m)$ is the mth error sample between the PA output and desired signal, and μ is a learning rate.

With the feature transformation learned via pretraining, we can calculate $\Delta\mathbf{w}$ with the LMS algorithm and then transform the coefficients back to $\Delta\mathbf{c}$. Thus, the LMS adaptation becomes

$$\mathbf{w^{(m+1)}} = \mathbf{w^{(m)}} - \mu\Delta\mathbf{w}_{\text{LMS}}^{\mathbf{(m)}}, \tag{39}$$

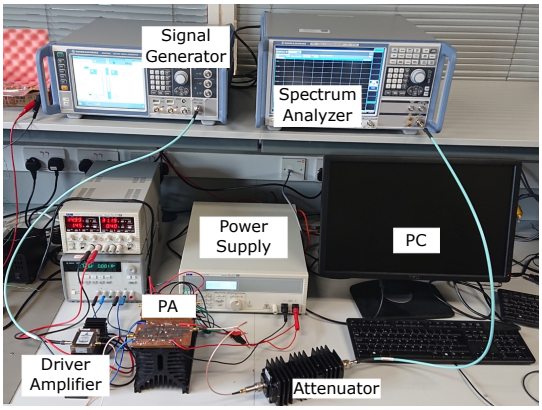


Fig. 6. The photograph of the DPD test bench.

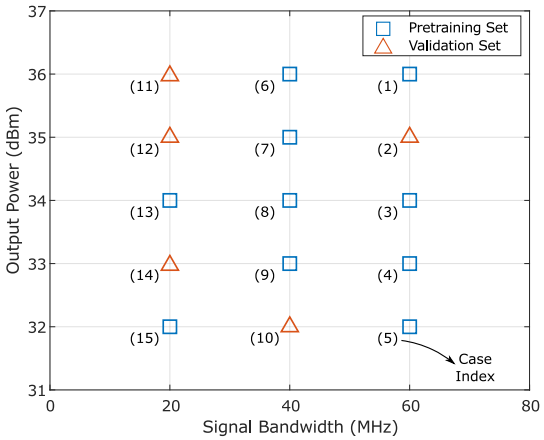


Fig. 7. Illustration of the pretraining and validation set.

where

$$\Delta \mathbf{w}_{\mathrm{LMS}}^{(\mathbf{m})} = \mathbf{A}^H \mathbf{X}(m)^H e(m), \tag{40}$$

and μ is the learning rate. The back-substitution can be done with (32).

The pretraining-assisted LMS algorithm can converge much faster than the conventional version, because fewer coefficients are updated and the original regression matrix $\mathbf{X}$ is made approximately orthogonal by $\mathbf{A}$. Nevertheless, due to the additional transformation operations, the per-iteration complexity is increased from Q to $(2P+1)Q$. To offset the undesired effect, in each iteration, we can choose $P_{\mathrm{LMS}} < P$ coefficients from $\mathbf{w}$ and only update the selected coefficients. Because of the orthogonality in $\mathbf{Z}$, fast convergence is still guaranteed. In this way, we can balance the per-iteration complexity and the convergence speed according to the specific application scenarios. The overall effect should be similar to the independent partial identification technique in [11].

IV. Results

A. Experimental Setup

To validate the proposed method, a test platform was set up as shown in Fig. 6, which included PC, signal generator, driver amplifier, PA, attenuator and spectrum analyzer. The PA under test was an in-house designed broadband GaN Doherty PA operating at 3.2 GHz with the maximum output power at 44 dBm. The sampling rate of baseband signals were set to 300 MSPS. The excitation input signals were carrier-aggregated LTE signals with 8 dB PAPR. 20,000 data samples were used in all tests. The generalized memory polynomial (GMP) model [33] was employed in DPD. Recorded I/Q input and output samples were time aligned and normalized before training the model. The time-alignment and model extraction were both performed in MATLAB.

In the experiments, the variations of power level and signal bandwidth were considered. The two quantities are chosen to best demonstrate the usefulness of our approach under the constraint of our instruments. While the power level can be continuously tuned, the signal bandwidth typically has limited choices and every change can potentially lead to large variations in PA behavior. Thus, the two quantities represent two different types of system factors. In addition, the change of signal bandwidth naturally involves different excitation signals, which helps verify the proposed generalized QR-SVD algorithm.

As a proof of concept, the average output power of PA could vary from 32 to 36 dBm with 1 dB step size and the available signal bandwidths were 20, 40 and 60 MHz. Thus, there were in total $5 \times 3 = 15$ possible configurations when we considered all combinations of bandwidth and power level.

To extract effective features, we conducted pretraining on 10 randomly selected cases and validated the performance on the rest 5 of them. We refer to the two sets as "pretraining set" and "validation set" thereafter. The selection was illustrated in Fig. 7. Note that in practical scenarios, the pretraining set can also be manually specified to include all representative cases. ILC algorithm [29] was performed on the selected cases to obtain the "ideal" DPD output signals required in pretraining.

B. Experimental Results With LS Adaptation

In the experimental tests, we used a GMP model with 64 coefficients. The model structure was selected such that good performance could be achieved under the most challenging case, i.e. 60 MHz signal bandwidth and 36 dBm output power. The selection was based on a sweep of model parameters in forward PA modeling, as is shown in Fig. 8. The same model structure was used by all cases throughout the experimental measurement section. It is worth mentioning that other Volterra series models or piecewise models may replace the GMP model trivially.

The first step of the proposed methodology was feature extraction which was performed offline. As more than one input signals were used, the generalized QR-SVD pretraining algorithm was used.

After performing pretraining, we obtained the relative importance of all principal components in Fig. 9. From the results of individually trained models, the baseline NMSE is approximately -43 dB. Thus, we set the threshold of principal component importance to be -53 dB, i.e., 10 dB below the performance baseline. According to Fig. 9, P was set to 6, which means the proposed method only identified 6 coefficients during DPD adaptation.

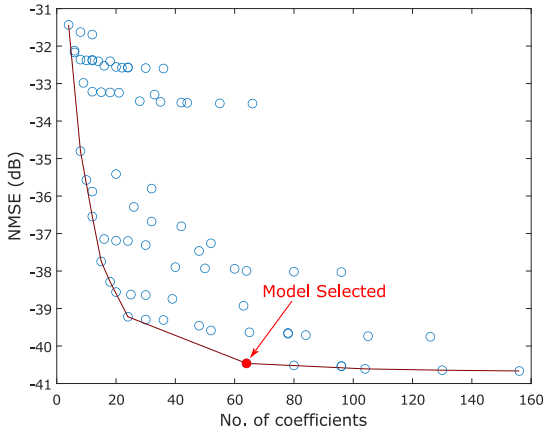


Fig. 8. Demonstration of DPD model selection.

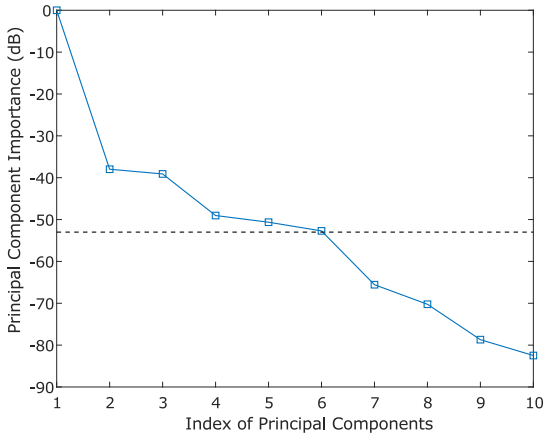


Fig. 9. Relative importance of principal components.

To verify the pretraining approach, a DPD test was performed. The DPD coefficients were iteratively updated using a closed-loop estimator. To draw a comparison, conventional adaptation method and the proposed pretraining-based adaptation were tested on all cases. The linearization performance of the two training methods under all test conditions is reported in Fig. 10, 11 and 12. From the measurement, the performance deviation is within 1 dB in all cases. It suggests the proposed adaptation procedure achieves comparable performance with the conventional method using a significantly lower number of effective coefficients. In addition, both pretraining and validation sets are well modeled, showing that the proposed pretraining method generalizes well to unseen configurations and can successfully extract effective features to model the varying PA behavior.

To better illustrate the DPD performance, we present more detailed results for selected operating conditions. Fig. 13 shows the spectral results for linearizing the PA driven at 36 dBm output power with a 60 MHz signal. It shows that the performance for the two DPD methods is very close. The AM-AM and AM-PM characteristics for the proposed pretraining method is depicted in Fig. 14.

To show the worst-case performance of the proposed method, we examine the case with the maximum performance deviation, i.e., 34 dBm output power with 60 MHz signal.

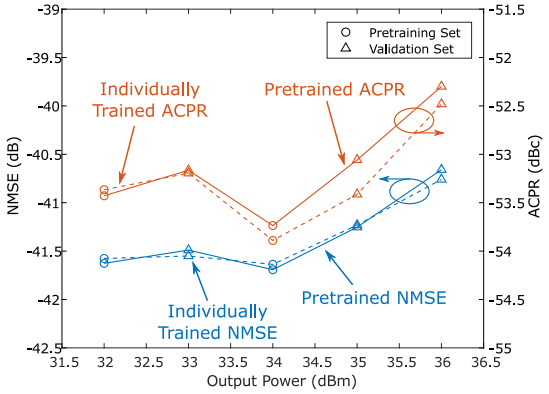


Fig. 10. DPD performance of individually trained and pretrained GMP models with 20 MHz signals.

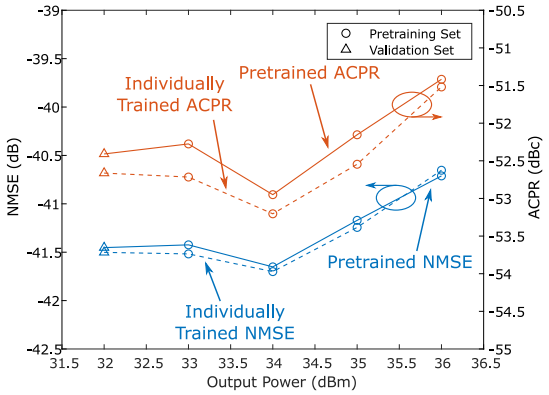


Fig. 11. DPD performance of individually trained and pretrained GMP models with 40 MHz signals.

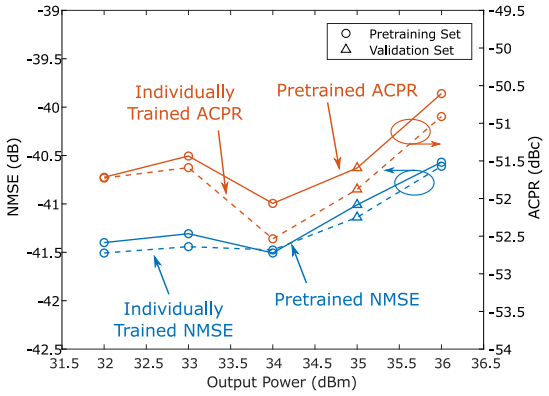


Fig. 12. DPD performance of individually trained and pretrained GMP models with 60 MHz signals.

From the spectral plot in Fig. 15, we can see that the performance of the two methods is still comparable. The AM-AM and AM-PM results of the proposed method are shown in Fig. 16.

Moreover, the proposed method also has the potential to reduce the required number of data samples for extraction. To validate this feature, we performed model adaptation using fewer data samples. The training samples were a continuous data segment which was randomly selected from the original training sequence. The sample size for training signals, N,

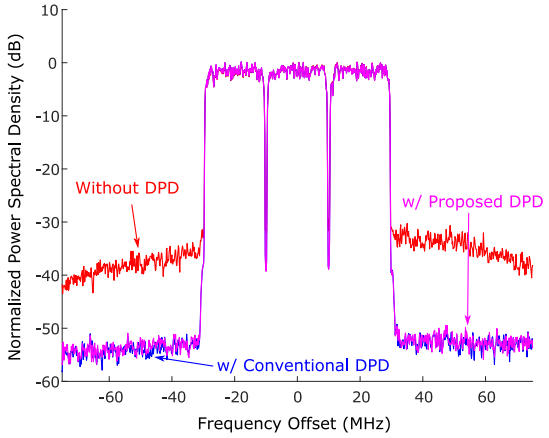


Fig. 13. Spectra of individually trained and pretrained DPD with 60 MHz signals and 36 dBm output power.

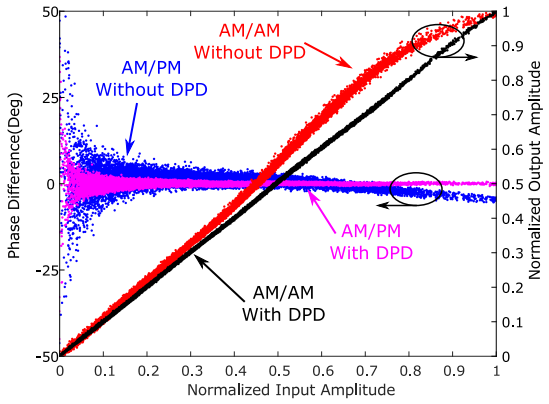


Fig. 14. AM-AM and AM-PM characteristics of pretrained DPD with 60 MHz signals and 36 dBm output power.

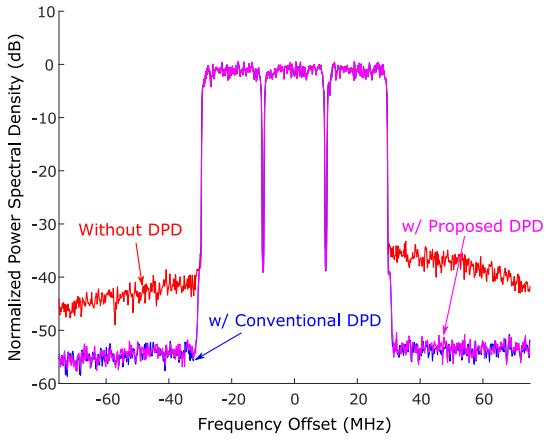


Fig. 15. Spectra of individually trained and pretrained DPD with 60 MHz signals and 34 dBm output power.

were changed in a set of measurements to quantify the effect. In the results shown in Fig. 17, the pretraining-assisted adaptation achieves robust performance even with as few as 50 training samples, while the same level of linearization performance requires more than 2000 samples in conventional adaptation solutions. Note that the conventional method would completely fail when the sample size is smaller than the

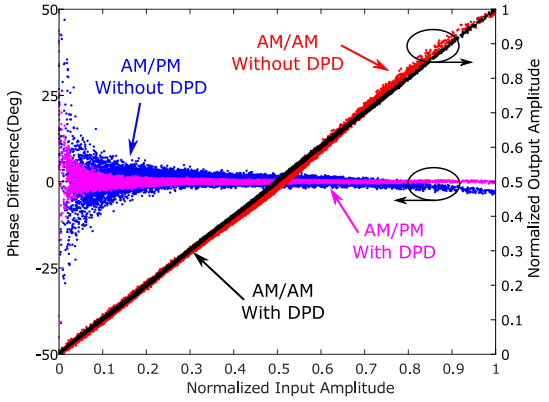


Fig. 16. AM-AM and AM-PM characteristics of pretrained DPD with 60 MHz signals and 34 dBm output power.

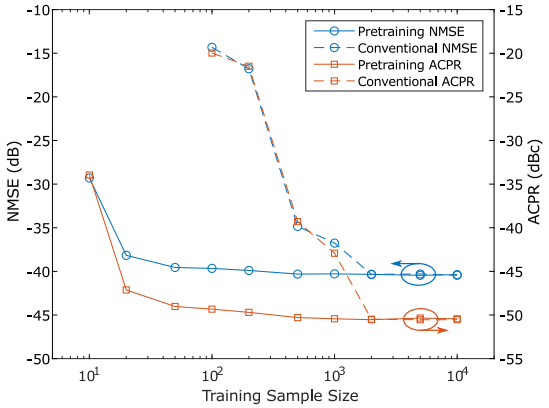


Fig. 17. DPD performance of pretrained models with varying training sample size.

number of model coefficients, and that is why the conventional method was not tested with sample sizes below 100.

We also experimentally verified the combination of the proposed pretraining method and CM-iDLA method in [9]. The DPD performance is depicted in Fig. 18. While the performance of the original CM-iDLA method is significantly degraded with a training sample size of 1000, its combination with the pretraining methodology can achieve reasonable performance with only 500 samples.

C. Simulation Results With LMS Adaptation

To run LMS based model extraction, a closed-loop real-time hardware implementation would be preferred because a large number of iterations is required in LMS adaptation. However, in our lab, we don't have such real-time hardware and thus it is difficult to run LMS in the experimental test. Instead, a simulation setup was used to verify the convergence performance in LMS algorithm. We extracted a PA model using the input and output data captured in the ILC test and used LMS algorithm to update the DPD coefficients. The PA model was a decomposed vector rotation (DVR) model [7] with 39 coefficients. The DPD model and the transformation matrix $\mathbf{A}$ were both the same as those used in the previous experimental tests.

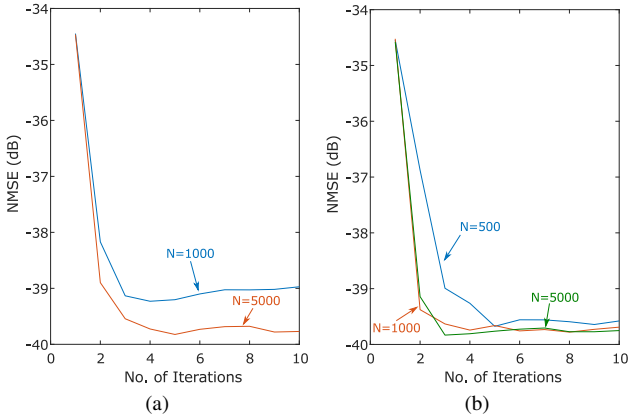


Fig. 18. DPD performance with CM-iDLA using different training sample size N: (a) conventional and (b) pretrained.

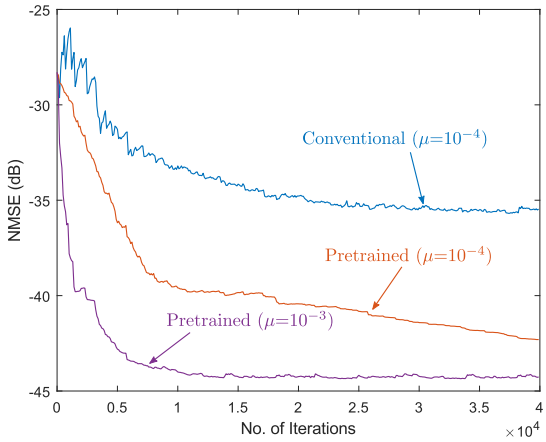


Fig. 19. DPD performance with LMS using different learning rate μ.

In the simulation, 40,000 data samples were used, which corresponds to 40,000 LMS iterations. For fair comparison, basis functions in the DPD model were all normalized with respect to their rooted mean square (RMS) value. Different learning rates were used during the simulation. While the pretrained LMS algorithm could converge with up to $\mu = 10^{-3}$, the conventional version diverged when learning rate was larger than $\mu = 10^{-4}$. It clearly shows that the proposed modification significantly improves the conditioning and robustness. The simulated NMSE performance was depicted in Fig. 19, showing dramatic improvement in convergence speed.

D. Complexity Analysis

In this section, we analyze the computational complexity of the proposed adaptation algorithms. We assume there are Q basis functions in the original DPD model and P features after pretraining. N_c training samples are used in conventional method and N_p samples in the proposed solution. Thus, in the GMP model tested in this work, we have $Q = 64, P = 6, N_c = 2000, N_p = 50$. The computational complexity is measured by the number of complex multiplications and the results are shown in Table I. The comparison shows significant reduction in computational complexity, as the number of multiplication is reduced to 0.47%, which is more than 200x reduction. Based on the analytical derivation, most reduction is due to the shortened training sequence (40x), and the reduced number of updated coefficients contribute to a further 5x reduction.

TABLE I
COMPLEXITY COMPARISON OF LS ADAPTATION

	Conventional	Proposed
$\mathbf{XA}$	0	PQN_p (19,200)
$\mathbf{X}^H\mathbf{X}$	Q^2N_c (4,160,000)	P^2N_p (1,050)
$(\mathbf{X}^H\mathbf{X})^{-1}$	Q^3 (262,144)	P^3 (216)
$\mathbf{X}^H\mathbf{y}$	QN_c (128,000)	PN_p (300)
$(\mathbf{X}^H\mathbf{X})^{-1}\mathbf{X}^H\mathbf{y}$	Q^2 (4,096)	P^2 (36)
$\mathbf{Aw}$	0	PQ (384)
Total	4,554,240 (100%)	21,186 (0.47%)

Note: The numbers are based on $Q = 64, P = 6, N_c = 2000, N_p = 50$

V. CONCLUSION

In this paper, we have presented a novel pretraining-assisted adaptation strategy to address the complexity issue in DPD model adaptation. When the PA nonlinearity is known to change within a certain range, we extract features that can model PA behaviors under all representative operating conditions using the proposed pretraining algorithms in an offline setup. After pretraining, the online model adaptation process only needs to update the transformed features, so the computational complexity can be greatly reduced. When applied to LMS-type algorithms, we demonstrate that the proposed method can significantly boost the convergence speed with moderate increase of complexity. Thus, the proposed pretraining framework is versatile and has the potential to generalize to different types of DPD models and adaptation algorithms, which helps overcome the new challenges in diverse DPD applications for future wireless communications.

REFERENCES

[1] J. Wood, *Behavioral Modeling and Linearization of RF Power Amplifiers*. Artech House, 2014.
[2] L. Guan and A. Zhu, "Green communications: Digital predistortion for wideband RF power amplifiers," *IEEE Microw. Mag.*, vol. 15, no. 7, pp. 84–99, Nov. 2014.
[3] M. Shafi *et al.*, "5G: A tutorial overview of standards, trials, challenges, deployment, and practice," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 6, pp. 1201–1221, Jun. 2017.
[4] J. Wood, "System-level design considerations for digital pre-distortion of wireless base station transmitters," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 5, pp. 1880–1890, May 2017.
[5] Y. Ma, Y. Yamao, Y. Akaiwa, and C. Yu, "FPGA implementation of adaptive digital predistorter with fast convergence rate and low complexity for multi-channel transmitters," *IEEE Trans. Microw. Theory Techn.*, vol. 61, no. 11, pp. 3961–3973, Nov. 2013.
[6] X. Wang, Y. Li, C. Yu, W. Hong, and A. Zhu, "Digital predistortion of 5G massive MIMO wireless transmitters based on indirect identification of power amplifier behavior with OTA tests," *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 1, pp. 316–328, Jan. 2020.
[7] A. Zhu, "Decomposed vector rotation-based behavioral modeling for digital predistortion of RF power amplifiers," *IEEE Trans. Microw. Theory Techn.*, vol. 63, no. 2, pp. 737–744, Feb. 2015.
[8] 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.

[9] 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.

[10] P. L. Gilabert *et al.*, "Order reduction of wideband digital predistorters using principal component analysis," in *2013 IEEE MTT-S International Microwave Symposium Digest (IMS)*, Jun. 2013, pp. 1–7.

[11] D. López-Bueno, Q. A. Pham, G. Montoro, and P. L. Gilabert, "Independent digital predistortion parameters estimation using adaptive principal component analysis," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 12, pp. 5771–5779, Dec. 2018.

[12] 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," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 10, pp. 3996–4004, Oct. 2019.

[13] Y. Li, W. Cao, and A. Zhu, "Instantaneous sample indexed magnitude-selective affine function-based behavioral model for digital predistortion of RF power amplifiers," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 11, pp. 5000–5010, Nov. 2018.

[14] 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," *IEEE Microw. Mag.*, vol. 20, no. 5, pp. 50–61, May 2019.

[15] 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," in *2007 Asia-Pacific Microwave Conference (APMC)*, Dec. 2007, pp. 1–4.

[16] P. L. Gilabert, G. Montoro, and E. Bertran, "FPGA implementation of a real-time NARMA-based digital adaptive predistorter," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 7, pp. 402–406, Jul. 2011.

[17] N. Kelly and A. Zhu, "Direct error-searching SPSA-based model extraction for digital predistortion of RF power amplifiers," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 3, pp. 1512–1523, Mar. 2018.

[18] N. Kelly and A. Zhu, "Low-complexity stochastic optimization-based model extraction for digital predistortion of RF power amplifiers," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 5, pp. 1373–1382, May 2016.

[19] O. Hammi, A. Kwan, and F. M. Ghannouchi, "Bandwidth and power scalable digital predistorter for compensating dynamic distortions in RF power amplifiers," *IEEE Trans. Broadcast.*, vol. 59, no. 3, pp. 520–527, Sep. 2013.

[20] R. N. Braithwaite, "A self-generating coefficient list for machine learning in RF power amplifiers using adaptive predistortion," in *2006 European Microwave Conference (EuMC)*, Sep. 2006, pp. 1229–1232.

[21] Y. Guo, C. Yu, and A. Zhu, "Power adaptive digital predistortion for wideband RF power amplifiers with dynamic power transmission," *IEEE Trans. Microw. Theory Techn.*, vol. 63, no. 11, pp. 3595–3607, Nov. 2015.

[22] A. S. Tehrani, T. Eriksson, and C. Fager, "Modeling of long term memory effects in RF power amplifiers with dynamic parameters," in *2012 IEEE/MTT-S International Microwave Symposium Digest (IMS)*, Jun. 2012, pp. 1–3.

[23] F. M. Barradas, L. C. Nunes, T. R. Cunha, P. M. Lavrador, P. M. Cabral, and J. C. Pedro, "Compensation of long-term memory effects on GaN HEMT-based power amplifiers," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 9, pp. 3379–3388, Sep. 2017.

[24] R. K. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *J. Mach. Learn. Res.*, vol. 6, pp. 1817–1853, Nov 2005.

[25] L. Ding *et al.*, "A robust digital baseband predistorter constructed using memory polynomials," *IEEE Trans. Commun.*, vol. 52, no. 1, pp. 159–165, Jan. 2004.

[26] F. M. Barradas, T. R. Cunha, and J. C. Pedro, "Digital predistortion of RF PAs for MIMO transmitters based on the equivalent load," in *2017 Integrated Nonlinear Microwave and Millimetre-wave Circuits Workshop (INMMiC)*, Apr. 20–21, 2017, pp. 1–4.

[27] P. N. Landin, S. Gustafsson, C. Fager, and T. Eriksson, "Weblab: A web-based setup for PA digital predistortion and characterization [application notes]," *IEEE Microw. Mag.*, vol. 16, no. 1, pp. 138–140, Feb. 2015.

[28] "RF WebLab - Online RF Measurements." [Online]. Available: http://dpdcompetition.com/rfweblab/.

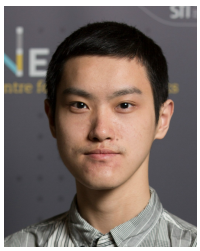
[29] J. Chani-Cahuana, P. N. Landin, C. Fager, and T. Eriksson, "Iterative learning control for RF power amplifier linearization," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 9, pp. 2778–2789, Sep. 2016.

[30] R. Raich, H. Qian, and G. T. Zhou, "Orthogonal polynomials for power amplifier modeling and predistorter design," *IEEE Trans. Veh. Technol.*, vol. 53, no. 5, pp. 1468–1479, Sep. 2004.

[31] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed., ser. Springer Series in Statistics. New York: Springer-Verlag, 2009.

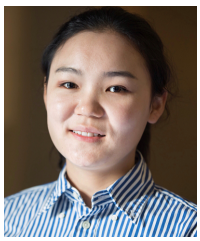
[32] R. N. Braithwaite, "Digital predistortion of an RF power amplifier using a reduced Volterra series model with a memory polynomial estimator," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 10, pp. 3613–3623, Oct. 2017.

[33] D. Morgan, Z. Ma, J. Kim, M. 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.



Yue Li (Member, IEEE) received the B.E. degree in information engineering from Southeast University, Nanjing, China, in 2016, and Ph.D. degree in electronic engineering from University College Dublin, Dublin, Ireland, in 2020.

He is currently a postdoctoral researcher with the RF and Microwave Research Group, UCD. His current research interests include behavioral modeling and digital predistortion for RF power amplifiers.



Xiaoyu Wang (Graduate Student Member, IEEE) received the B.E. degree in information engineering from Southeast University, Nanjing, China, in 2015. She is currently working towards the Ph.D. degree at University College Dublin, Dublin, Ireland.

She is currently with the RF and Microwave Research Group, UCD. Her current research focuses on digital predistortion for RF power amplifiers, with a particular emphasis on applications to multiple-input multiple-output (MIMO) systems.

Anding Zhu (Senior Member, IEEE) received the Ph.D. degree in electronic engineering from University College Dublin (UCD), Dublin, Ireland, in 2004.

He is currently a Professor with the School of Electrical and Electronic Engineering, UCD. His research interests include high-frequency nonlinear system modeling and device characterization techniques, high-efficiency power amplifier design, wireless transmitter architectures, digital signal processing, and nonlinear system identification algorithms. He has published more than 130 peer-reviewed journal and conference articles.

Prof. Zhu is an elected member of MTT-S AdCom, the Chair of the Electronic Information Committee and the Vice Chair of the Publications Committee. He is also the Chair of the MTT-S Microwave High-Power Techniques Committee. He served as the Secretary of MTT-S AdCom in 2018. He was the General Chair of the 2018 IEEE MTT-S International Microwave Workshop Series on 5G Hardware and System Technologies (IMWS-5G) and a Guest Editor of the IEEE Transactions on Microwave Theory and Techniques on 5G Hardware and System Technologies. He is currently an Associate Editor of the IEEE Microwave Magazine and a Track Editor of the IEEE Transactions on Microwave Theory and Techniques.