Iterative Multimetric Model Extraction for Digital Predistortion of RF Power Amplifiers Using Enhanced Quadratic SPSA

Hang Yin, *Graduate Student Member, IEEE*, and Anding Zhu, *Fellow, IEEE*

***Abstract*— In upcoming wireless systems, different linearity requirements in different frequency regions may be required due to data load and channel condition differences. This article presents a novel approach for identifying digital predistortion (DPD) coefficients that target multimetric linearity performance. First, a mathematical framework is established to evaluate the multimetric linearity performance in different frequency regions. Then, the DPD coefficients are extracted by optimizing the multimetric evaluation using the iterative quadratic simultaneous perturbation stochastic approximation (Q-SPSA) algorithm, which has been enhanced for improved numerical stability and reduced complexity. The experimental results demonstrate that the proposed approach achieves the multimetric requirements with fewer computational resources compared to conventional methods that only optimize the time-domain normalized mean square error (NMSE) metric. The proposed approach sheds light on DPD models, showing that DPD coefficients can be optimized to have different multimetric performances as the iteration process progresses. Detailed experimental results and complexity analysis further support the potential of the proposed approach for future systems.**

***Index Terms*— Digital predistortion (DPD), multiobjective optimization, power amplifier (PA), simultaneous perturbation stochastic optimization (SPSA).**

I. Introduction

THE advent of next-generation wireless systems, such as 6G and Wi-Fi 7, will bring further improvements in data transmission rate and lower latency through the implementation of advanced features and techniques [1], [2]. For instance, Wi-Fi 7 may utilize high-order modulation techniques of up to 4096-quadrature amplitude modulation (QAM) and a wider transmission bandwidth of 320 MHz [3], [4]. To accommodate these enhancements, digital predistortion (DPD) is gaining importance in both cellular base stations [5], [6], [7] and Wi-Fi transmitters [8], [9], [10] as it ensures that power amplifiers (PAs) can operate with both high efficiency and high linearity.

One of the key characteristics of future wireless systems is coordination. For instance, in Wi-Fi 7, multiple access points (APs) can work together in a collaborative manner to efficiently allocate communication resources [11], [12], [13], [14]. As a result, different linearity requirements can arise in different frequency regions based on the coordination scenario. For example, the in-band (IB) linearity requirement, closely related to the error vector magnitude (EVM) indicator, can vary in strictness depending on the deployed modulation order, which is influenced by the distribution of data load and channel condition. The out-of-band (OOB) linearity requirement, such as the spectrum mask requirement, can also fluctuate depending on channel conflicts during coordination among multiple APs. As depicted in Fig. 1, for a specific AP or wireless transmitter, its nonlinearity may need to be suppressed to varying levels in different frequency regions. The solid curve represents the original distortion level distribution in the frequency domain without introducing DPD, while the dashed line shows the required linearity error levels after DPD in different regions. In some bands, such as Band-A and Band-B, the required linearity level may be relaxed when the OOB regions are lightly occupied or not occupied, while in Band-E and Band-F, where the OOB is heavily occupied, the distortion must be suppressed to lower levels to avoid impacting other APs using the channel. In IB regions, that is, Band-C and Band-D, higher-order QAM also requires lower linearity error levels. This presents significant challenges for DPD operation since in the existing systems, DPD is operated in the time domain, which does not distinguish distortions in different frequency bands.

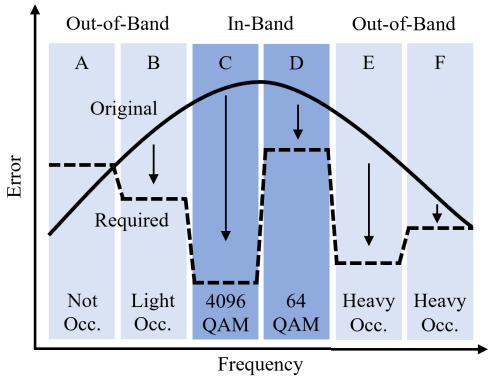


Fig. 1. Multimetric linearity requirements.

Manuscript received 28 April 2023; revised 11 July 2023 and 6 August 2023; accepted 8 August 2023. This work was supported by the Science Foundation Ireland under Grant 16/IA/4449. *(Corresponding author: Hang Yin.)*

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

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMTT.2023.3305181.

Digital Object Identifier 10.1109/TMTT.2023.3305181

Furthermore, the existing DPD usually uses the least-square (LS) algorithm to extract coefficients, particularly, for the linear-in-parameter models such as memory polynomial (MP) [15], generalized MP (GMP) [16], dynamic deviation-reduction (DDR) [17], and decomposed-vector rotation (DVR) [18]. The LS is a closed-form solution and can find the optimal coefficients in just one iteration. However, very high computational complexity is involved that can put a heavy burden on hardware resources. Moreover, in many cases, a suboptimal solution might be sufficient to satisfy the linearity requirement, as opposed to the optimal solution found by LS. To reduce computational complexity, iterative algorithms, such as the quadratic simultaneous perturbation stochastic approximation (Q-SPSA), which updates the coefficients by random perturbations on coefficients with low algorithmic and hardware implementation complexity, is of growing interest in DPD recently [19], [20], [21]. However, the existing Q-SPSA algorithms extract the DPD coefficients via minimizing the time-domain normalized mean square error (NMSE) only, which cannot approach linearity requirements in different frequency regions in a computationally cost-effective manner.

Concerning achieving different linearization performances in different frequency regions, there are some methods proposed in the literature. In [22], a band-limited DPD concept was proposed that uses a filter to control the linearization bandwidth in the system. Later in [23], the concept was generalized to frequency-selective DPD in which the focused band of linearization can be selected not only for IB, but also for OOB. Both [22] and [23] have to use a filter at the output of PA to process the signals in different frequency regions. This implies that the passband of the filter must be modified when the band of interest changes from time to time, which could undermine the flexibility of the system. Besides, these methods still use LS to extract the DPD coefficients for the interested band, which suffers high computational complexity.

In this article, a multimetric DPD concept is proposed and a low-complexity iterative model extraction algorithm based on Q-SPSA is derived. A mathematical form of multimetric linearity evaluation for characterizing linearity error in different frequency regions is first established and then the DPD coefficients are identified based on the multimetric evaluation. When compared with our preliminary work [24], we provide not only more details in the algorithm derivation, but also conduct detailed complexity analysis and make it applicable in practical online DPD model adaptation scenarios. An enhanced way of Q-SPSA progression step calculation is also proposed and multiple experimental validations are given.

The rest of this article is organized as follows. Section II will further introduce theoretical foundations for multimetric DPD. Section III will introduce the DPD implementation details consisting of DPD system architecture, specific algorithm operations, as well as complexity analysis. Then experimental results will be given in Section IV. Finally, Section V will be the conclusion.

II. Theoretical Foundations

In this section, we will introduce the theoretical foundation of multimetric linearization. We will start by discussing the conventional single-metric NMSE-based optimization and then move on to explain how it is derived to become multimetric linearization and the introduction of a new loss function. All of these are established based on simple and neat algebraic-level operations.

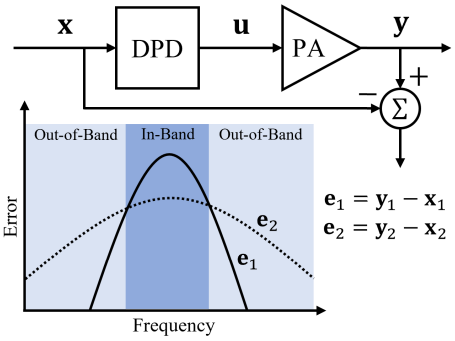


Fig. 2. DPD schematic and error spectrum comparison.

A. Single-Metric Linearization

In conventional DPD model extraction, either using LS [15], [16], [18] or SPSA [19], [20], [21], the optimization target is NMSE, that is the normalized sum of the linearity errors between the desired transmitted signal $\mathbf{x}$ and the distorted PA output signal $\mathbf{y}$, as shown in Fig. 2, and it can be expressed as follows:

$$\mathrm{NMSE}(\mathbf{x},\mathbf{y}) = 10\log_{10}\frac{(\mathbf{y}-\mathbf{x})^H(\mathbf{y}-\mathbf{x})}{\mathbf{x}^H\mathbf{x}}\ (\mathrm{dB}). \tag{1}$$

For simplicity, if we define linearity error vector as $\mathbf{e}=\mathbf{y}-\mathbf{x}$ and ignore the constant denominator $\mathbf{x}^H\mathbf{x}$ that is used for normalization, the sum of square error (SSE), expressed as follows:

$$L(\mathbf{e}) = \mathrm{SSE}(\mathbf{x},\mathbf{y}) = \mathbf{e}^H\mathbf{e} \tag{2}$$

can be used as an equivalent indicator for NMSE. It is not hard to find that the SSE indicator is a scalar that counts the energy of error vector $\mathbf{e}$ in the time domain only, which means that, at the same SSE level, that is,

$$\mathbf{e}_1^H\mathbf{e}_1 = \mathbf{e}_2^H\mathbf{e}_2 \tag{3}$$

the linearity error distribution in different frequency regions, such as IB and OOB, could be totally different, as illustrated in Fig. 2, where the solid and dashed curves are the spectrums for $\mathbf{e}_1$ and $\mathbf{e}_2$, respectively, with $\mathbf{e}_1$ has more IB energy, while $\mathbf{e}_2$ has more OOB energy.

Since the error is not distinguished in the frequency domain, the NMSE- or SSE-based model extraction will reduce the linearity error blindly in all frequency regions. In other words, if using the NMSE or SSE as the only target, the linearity error in different frequency regions, such as IB and OOB errors, cannot be reduced with the desired preference. The conventional DPD, therefore, cannot effectively achieve the satisfaction of different linearity requirements in different frequency regions, as shown in Fig. 1.

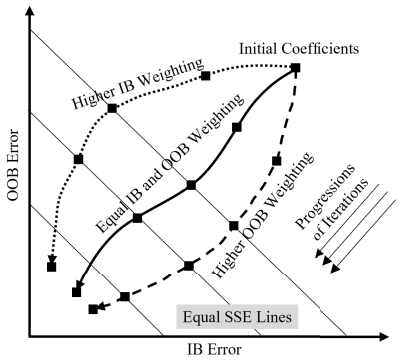


Fig. 3. Different performance trajectories.

B. Multimetric Linearization

In future systems, it is desirable to have different linearity requirements in different frequency regions such as IB and OOB. Thus, IB and OOB errors should be optimized with preference. Fig. 3 illustrates different performance trajectories in terms of IB and OOB errors with progression of iteration, where the middle solid trajectory is assumed to have equal weightings on IB and OOB, and by placing different weightings on IB and OOB, different performance trajectories can be obtained, for example, with higher IB weighting, the upper dashed trajectory in Fig. 3 allows a higher OOB error but a lower IB error, at the same error level of SSE.

To characterize linearity errors in different frequency regions, new measurements of $\mathbf{e}$ must be established. An indicator of the sum of channel power (SCP) proposed in our earlier work [24] can be used to calculate the energy of error signal $\mathbf{e}$ in different channels, as follows:

$$\mathrm{SCP}(\mathbf{e}) = \mathbf{e}^H \mathbf{F}^H \mathbf{B} \mathbf{F} \mathbf{e} \tag{4}$$

where $\mathbf{e}$ is of size $N \times 1$ and N is the number of signal samples, and $\mathbf{F}$ is the normalized N-point discrete Fourier transform (DFT) matrix (with zero frequency locating in center positions), and $\mathbf{B}$ is the band-selection matrix of $N \times N$ that selects the band for summing up energy. In detail, $\mathbf{B}$ is diagonal with filling in its diagonal positions to select corresponding DFT frequency components. For example, to compute center-band or IB energy, $\mathbf{B}$ will be

$$\mathbf{B} = \begin{bmatrix} \mathbf{O}_{(N-n)/2} & & \\ & \mathbf{E}_n & \\ & & \mathbf{O}_{(N-n)/2} \end{bmatrix} \tag{5}$$

where $\mathbf{E}_n$ is an identity matrix with size n, which is the number of discrete DFT frequency points n that corresponds to the modulated signal bandwidth. $\mathbf{O}_{(N-n)/2}$ is a zero matrix with size $(N-n)/2$. When $\mathbf{B}$ is diagonally fully filled as an identity matrix, the SCP will compute full-band energy, and its expression will degenerate to SSE in (2) due to the unitary property of DFT matrix such that

$$\mathbf{F}^H \mathbf{F} = \mathbf{E}. \tag{6}$$

To sum up energy in several frequency regions, different diagonal positions in $\mathbf{B}$ can be filled correspondingly, as illustrated in Fig. 4, where the energy in three different frequency regions

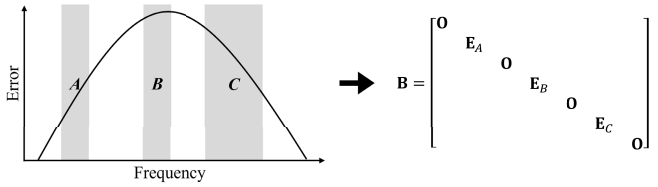


Fig. 4. SCP and corresponding band-selection matrix.

is summed up. The SCP indicator can also be used separately with $\mathbf{B}$ of different fillings, for example,

$$\begin{aligned} \mathrm{SCP}_1(\mathbf{e}) &= \mathbf{e}^H \mathbf{F}^H \mathbf{B}_1 \mathbf{F} \mathbf{e} \\ \mathrm{SCP}_2(\mathbf{e}) &= \mathbf{e}^H \mathbf{F}^H \mathbf{B}_2 \mathbf{F} \mathbf{e} \\ \mathrm{SCP}_3(\mathbf{e}) &= \mathbf{e}^H \mathbf{F}^H \mathbf{B}_3 \mathbf{F} \mathbf{e} \\ \mathrm{SCP}_4(\mathbf{e}) &= \mathbf{e}^H \mathbf{F}^H \mathbf{B}_4 \mathbf{F} \mathbf{e}. \end{aligned} \tag{7}$$

To avoid confusion, it is worth clarifying that the band selections here that we refer to, for example, the A, B, and C bands illustrated in Fig. 4, are the different frequency regions within the single bandwidth of the PA. In other words, the selected frequency regions of interest are around one carrier frequency.

Without loss of generality and for ease of explanation and visualization, this article will mainly focus on two typical frequency regions of IB and OOB to introduce and validate the idea. To calculate IB and OOB energy, $\mathbf{B}_1$ will be the same as (5) and $\mathbf{B}_2 = \mathbf{E} - \mathbf{B}_1$ will be

$$\mathbf{B}_2 = \begin{bmatrix} \mathbf{E}_{(N-n)/2} & & \\ & \mathbf{O}_n & \\ & & \mathbf{E}_{(N-n)/2} \end{bmatrix}. \tag{8}$$

Similar to the mathematical form in (4), the linearity vector $\mathbf{e}$ can be decomposed into IB and OOB components as follows:

$$\mathbf{e} = \mathbf{e}_{\mathrm{IB}} + \mathbf{e}_{\mathrm{OOB}} \tag{9}$$

$$= \mathbf{F}^H \mathbf{B}_1 \mathbf{F} \mathbf{e} + \mathbf{F}^H \mathbf{B}_2 \mathbf{F} \mathbf{e}. \tag{10}$$

The two $\mathbf{F}^H$s will transform the selected frequency components back into the time domain. Substituting the specific $\mathbf{B}_1$ and $\mathbf{B}_2$ and unitary property (6) into (10), it is not hard to find it holds. Equations (9) and (10) is just decomposition of $\mathbf{e}$ without preference, and to add in preferences, a new error vector can be generated as follows:

$$\epsilon = \alpha \mathbf{F}^H \mathbf{B}_1 \mathbf{F} \mathbf{e} + (1-\alpha) \mathbf{F}^H \mathbf{B}_2 \mathbf{F} \mathbf{e} \tag{11}$$

where $\alpha \in [0, 1]$ is a real number that controls the relative contribution of IB and OOB components, and when $\alpha = 0.5$, ϵ is with equal preference on IB and OOB errors and will degenerate to $\mathbf{e}/2$. The derived new loss function for multimetric linearization will be

$$\begin{aligned} L(\mathbf{e}) &= \epsilon^H \epsilon \\ &= \alpha^2 \mathbf{e}^H \mathbf{F}^H \mathbf{B}_1 \mathbf{F} \mathbf{e} + (1-\alpha)^2 \mathbf{e}^H \mathbf{F}^H \mathbf{B}_2 \mathbf{F} \mathbf{e} \\ &= \mathbf{e}^H \mathbf{F}^H \mathbf{\Psi} \mathbf{F} \mathbf{e} \end{aligned} \tag{12}$$

where

$$\mathbf{\Psi} = \alpha^2 \mathbf{B}_1 + (1-\alpha)^2 \mathbf{B}_2 \tag{13}$$

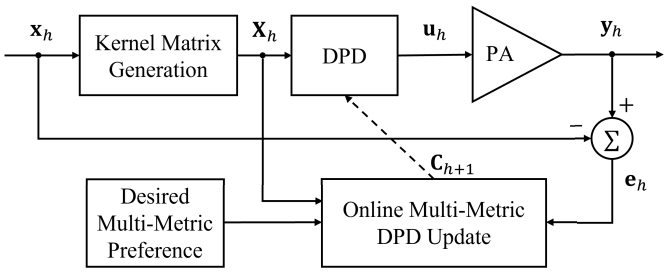


Fig. 5. Multimetric DPD system architecture.

and it can be found that the weightings on $\mathbf{e}_{\text{IB}}$ and $\mathbf{e}_{\text{OOB}}$ result in squared effect on two metrics of IB error $\mathbf{e}^H\mathbf{F}^H\mathbf{B}_1\mathbf{F}\mathbf{e}$ and OOB error $\mathbf{e}^H\mathbf{F}^H\mathbf{B}_2\mathbf{F}\mathbf{e}$.

For scenarios where multiple frequency divisions, rather than IB and OOB divisions are required, the loss function can be established as follows:

$$\begin{aligned} L(\mathbf{e}) &= \sum_i \alpha_i^2 \mathbf{e}^H\mathbf{F}^H\mathbf{B}_i\mathbf{F}\mathbf{e} \\ &= \mathbf{e}^H\mathbf{F}^H\boldsymbol{\Psi}\mathbf{F}\mathbf{e} \end{aligned} \tag{14}$$

with

$$\boldsymbol{\Psi} = \sum_i \alpha_i^2 \mathbf{B}_i \tag{15}$$

where $\mathbf{B}_i$ designates specific frequency divisions and α_i^2 is the corresponding weighting. Note that a squared form is used to keep it similar to (13), and it is a more general case when compared with (13).

III. DPD Implementation

A. System Architecture

The entire architecture of the proposed multimetric DPD system is given in Fig. 5, where $\mathbf{x}_h$ represents the modulated signal sequence to be transmitted at iteration index h. In the online DPD identification process, each $\mathbf{x}_h$ at iteration h will be different as it is generated from real-time data bit streams. $\mathbf{X}_h$ is the model kernel matrix generated by $\mathbf{x}_h$, for example, using MP [15] or GMP [16] terms. $\mathbf{u}_h$ is the predistorted signal generated by $\mathbf{X}_h$ and current DPD coefficients $\mathbf{C}_h$, and $\mathbf{y}_h$ is the corresponding PA output signal. The online multimetric DPD update will be conducted using the linearity error vector $\mathbf{e}_h = \mathbf{y}_h - \mathbf{x}_h$, kernel matrix $\mathbf{X}_h$, and desired multimetric preference on linearity error in different frequency regions. Then new DPD coefficients $\mathbf{C}_{h+1}$ will be generated and used as the "old" coefficients for the next iteration $h+1$. Details will be given in Sections III-B and III-C. We will first brief the Q-SPSA algorithm for the direct learning DPD. After that, the Q-SPSA algorithm will be derived for multimetric DPD identification, and finally, an enhanced method to calculate the progression step will be given.

B. Online DPD Update Using Q-SPSA

In direct learning DPD, it is assumed that the linearity error after the PA, that is, $\mathbf{e}_h = \mathbf{y}_h - \mathbf{x}_h$, can be linearly backpropagated before the PA [20], [25], [26] and then $-\mathbf{e}_h$ is regarded as the amount of DPD output update, that is,

$$\mathbf{u}_{h+1} - \mathbf{u}_h \approx -\mathbf{e}_h. \tag{16}$$

For linear-in-parameter DPD models, the above equation can be further expressed as follows:

$$\mathbf{X}\mathbf{C}_{h+1} - \mathbf{X}\mathbf{C}_h = \mathbf{X}\boldsymbol{\Delta}\mathbf{C} \approx -\mathbf{e}_h \tag{17}$$

which means that it is assumed that an update of $\boldsymbol{\Delta}\mathbf{C}$ on the DPD coefficients will make the PA output updated by $\mathbf{X}\boldsymbol{\Delta}\mathbf{C}$. To eliminate $\mathbf{e}_h$, that is, make $\mathbf{y}_h$ and $\mathbf{x}_h$ as close as possible, $\mathbf{X}\boldsymbol{\Delta}\mathbf{C}$ and $-\mathbf{e}_h$ should then be as close as possible. Thus, in direct learning DPD, the actual residual vector to be evaluated is

$$\mathbf{r}_h = \mathbf{e}_h + \mathbf{X}\boldsymbol{\Delta}\mathbf{C}. \tag{18}$$

In the single-metric SSE optimization, error vector evaluation (2) is employed and the actual loss function for acquiring $\boldsymbol{\Delta}\mathbf{C}$ will be

$$\begin{aligned} L(\mathbf{r}_h) &= L(\mathbf{e}_h + \mathbf{X}\boldsymbol{\Delta}\mathbf{C}) \\ &= (\mathbf{e}_h + \mathbf{X}\boldsymbol{\Delta}\mathbf{C})^H(\mathbf{e}_h + \mathbf{X}\boldsymbol{\Delta}\mathbf{C}) \\ &= L(\boldsymbol{\Delta}\mathbf{C}). \end{aligned} \tag{19}$$

The objective can be realized by LS as follows:

$$\boldsymbol{\Delta}\mathbf{C}_{\text{LS}} = -\left(\mathbf{X}_h^H\mathbf{X}_h\right)^{-1}\mathbf{X}_h^H\mathbf{e}_h \tag{20}$$

which will be of great complexity [19], [20] and not favorable for online DPD scenario as the above operations may not be conducted in a timely manner before the coming of following $\mathbf{X}_{h+1}$ and $\mathbf{e}_{h+1}$. As opposed to the above huge resource-consuming operations, the Q-SPSA algorithm was proposed in [20] to realize the minimization of (19) by three loss function measurements

$$\begin{aligned} L(+\boldsymbol{\Delta}_h) &= (\mathbf{e}_h + \mathbf{X}_h\boldsymbol{\Delta}_h)^H(\mathbf{e}_h + \mathbf{X}_h\boldsymbol{\Delta}_h) \\ L(-\boldsymbol{\Delta}_h) &= (\mathbf{e}_h - \mathbf{X}_h\boldsymbol{\Delta}_h)^H(\mathbf{e}_h - \mathbf{X}_h\boldsymbol{\Delta}_h) \\ L(\mathbf{0}) &= \mathbf{e}_h^H\mathbf{e}_h \end{aligned} \tag{21}$$

at $\boldsymbol{\Delta}\mathbf{C} = \mathbf{0}$ (zero vector) and $\pm\boldsymbol{\Delta}_h$, corresponding to not perturbing the coefficients and perturbing the coefficients by $\pm\boldsymbol{\Delta}_h$, where $\boldsymbol{\Delta}_h$ is the current perturbation vector which is of the same size as coefficients vector $\mathbf{C}_h$. Then the update of the coefficients vector will be found by

$$\boldsymbol{\Delta}\mathbf{C}_{\text{Q-SPSA}} = -\frac{L(+\boldsymbol{\Delta}_h) - L(-\boldsymbol{\Delta}_h)}{2\big(L(+\boldsymbol{\Delta}_h) - 2L(\mathbf{0}) + L(-\boldsymbol{\Delta}_h)\big)}\boldsymbol{\Delta}_h. \tag{22}$$

An illustration of $\boldsymbol{\Delta}\mathbf{C}$ obtaining at iteration h using LS and Q-SPSA is given in Fig. 6, where the LS gets the global optimal solution that minimizes (19) while the Q-SPSA algorithm gets the optimal solution in the quadratic slice determined by $\mathbf{0}$ (zero vector) and perturbation direction $\pm\boldsymbol{\Delta}_h$, which means that the obtained $\boldsymbol{\Delta}\mathbf{C}_{\text{Q-SPSA}}$ must be in the straight line that passes $\mathbf{0}$ and $\pm\boldsymbol{\Delta}_h$. For online DPD identification, the Q-SPSA will bring about two advantages. One is that the complexity of (22) is much lower than that of (20) [19], [20], and the other one is that (22) is usually smaller in scale when compared with (20), which will make the direct learning assumptions (16) and (17) better hold.

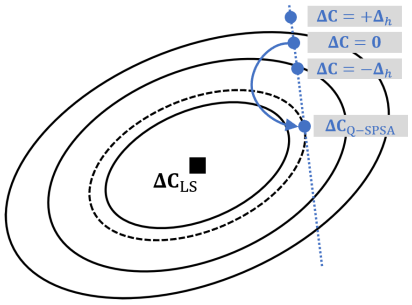


Fig. 6. Different algorithms to acquire $\mathbf{\Delta C}$ at iteration h.

C. Multimetric DPD Update Using Enhanced Q-SPSA

To realize multimetric DPD identification, (12) will be employed to conduct error evaluation of direct learning residual vector $\mathbf{r}_h$. Similar to (21), three new loss functions can be derived as follows:

$$\begin{aligned} L(+\mathbf{\Delta}_h) &= (\mathbf{e}_h + \mathbf{X}_h\mathbf{\Delta}_h)^H\mathbf{F}^H\mathbf{\Psi}\mathbf{F}(\mathbf{e}_h + \mathbf{X}_h\mathbf{\Delta}_h) \\ L(-\mathbf{\Delta}_h) &= (\mathbf{e}_h - \mathbf{X}_h\mathbf{\Delta}_h)^H\mathbf{F}^H\mathbf{\Psi}\mathbf{F}(\mathbf{e}_h - \mathbf{X}_h\mathbf{\Delta}_h) \\ L(\mathbf{0}) &= \mathbf{e}_h^H\mathbf{F}^H\mathbf{\Psi}\mathbf{F}\mathbf{e}_h. \end{aligned} \tag{23}$$

Equation (22) can also be employed to calculate the desired coefficients update under the above three loss measurements for multimetric identification, and the full progression step of DPD coefficients update will be

$$\mathbf{C}_{h+1} = \mathbf{C}_h + \mathbf{\Delta C}_{\mathrm{Q-SPSA}} \tag{24}$$

but it is found that $\mathbf{\Delta C}_{\mathrm{Q-SPSA}}$ can be calculated in an enhanced way that is mathematically equivalent and with many advantages. The enhanced progression step would be

$$\mathbf{C}_{h+1} = \mathbf{C}_h - \frac{R\{\mathbf{\Delta}_h^H\mathbf{X}_h^H\mathbf{F}^H\mathbf{\Psi}\mathbf{F}\mathbf{e}_h\}}{\mathbf{\Delta}_h^H\mathbf{X}_h^H\mathbf{F}^H\mathbf{\Psi}\mathbf{F}\mathbf{X}_h\mathbf{\Delta}_h}\mathbf{\Delta}_h \tag{25}$$

where $R\{\cdot\}$ is the real part operator. Detailed derivations are given in Appendix A and the complexity reduction will be discussed in Section III-D. In addition, this enhanced progression step will also bring better numerical stability. Specifically, in later iterations, $L(+\mathbf{\Delta}_h)$, $L(-\mathbf{\Delta}_h)$, and $L(\mathbf{0})$ could be very close, and in the conventional step (22) and (24), catastrophic cancellation effect [27] could be introduced, in which precision will be lost when adding/subtracting these very close values. In comparison, the enhanced step (25) avoids adding/subtracting these close values. The complete DPD implementation procedure of the proposed method is given in Algorithm 1.

D. Complexity Analysis

Now let us discuss the complexity of the proposed multimetric DPD iteration step and related comparisons. First, when compared with closed-form LS identification which is resource extensive in terms of both algorithmic level and hardware resources, the Q-SPSA, as an iterative identification algorithm, is of very low per iteration complexity [19], [20] and flexible to find adequate solutions of DPD coefficients that just satisfy the requirements to realize performance versus complexity tradeoff. The detailed complexity analysis and comparison will be conducted in two parts. The first part will analyze the complexity of the existing Q-SPSA progression step in (24) with (22) and the enhanced step in (25). In the second part, we will compare the complexity of the enhanced Q-SPSA for single-metric SSE and multimetric optimization, to provide a more reasonable comparison that considers not only the number of iterations, but also the complexity per iteration.

Algorithm 1 Online Multimetric DPD Model Extraction

Input: H different input sequences $\{\mathbf{x}_h\}$, $h = 1, 2, \ldots, H$, initial coefficient vector $\mathbf{C}_1$, desired multimetric preference parameter α

Output: DPD coefficient vector $\mathbf{C}_h$

1: **initialization**
Iteration index $h \leftarrow 1$
2: **repeat**
3: Generate model kernel matrix $\mathbf{X}_h$ from $\mathbf{x}_h$
4: Generate predistorted signal $\mathbf{u}_h = \mathbf{X}_h\mathbf{C}_h$
5: Send $\mathbf{u}_h$ to PA, and Collect PA output $\mathbf{y}_h$
6: Generate $\mathbf{C}_{h+1}$ from $\mathbf{C}_h$ by (25)
7: $h \leftarrow h + 1$
8: **until** IB and OOB error requirements are satisfied **or** iteration number limit is reached: $h > H$
9: **return** $\mathbf{C}_h$

1) Existing and Enhanced Q-SPSA for Multimetric Optimization: The complexity analysis is based on floating-point operations (FLOPS). The overall FLOPS consumed for each conventional Q-SPSA [20] iteration step in (24) with (22) and enhanced Q-SPSA step in (25), when applied on the same multimetric loss function (12), is provided as follows. The FLOPS for conventional Q-SPSA step on the multimetric loss function is

$$\beta_1 = N(4K + 15\log_2 N - 8) + 4K + 39 \tag{26}$$

and the FLOPS for enhanced Q-SPSA step on the multimetric loss function is

$$\beta_2 = N(4K + 10\log_2 N - 8) + 4K + 23 \tag{27}$$

where details to derive these values can be found in Appendix B. The enhanced method enjoys a complexity reduction of $5N\log_2 N + 16$ FLOPS per iteration step. As mentioned earlier, the existing step (24) with (22) and the enhanced step (25) are actually mathematically equivalent in which the savings are from the algebraic level calculations as detailed in Appendix A. When there are $K = 50$ coefficients and $N = 16\,384$ samples, the enhanced step consumes 82.6% FLOPS of the conventional step. Besides, the proposed step can also avoid the catastrophic cancellation effect as introduced earlier. Thus, the proposed enhanced Q-SPSA step (25) will be used as the common basis for both multimetric and single-metric SSE optimization comparison.

2) Enhanced Q-SPSA for Multimetric and Single-Metric SSE Optimization: In our preliminary work [24], it is demonstrated that using the multimetric loss function will result in a significantly smaller number of iterations to reach some multimetric requirements, when compared with the case that SSE-only one-metric is used as the loss function. However, not only the number of iterations, but also the complexity of

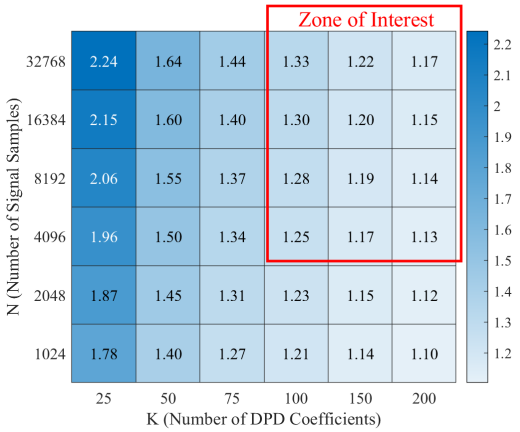


Fig. 7. Complexity ratio γ for different K and N values.

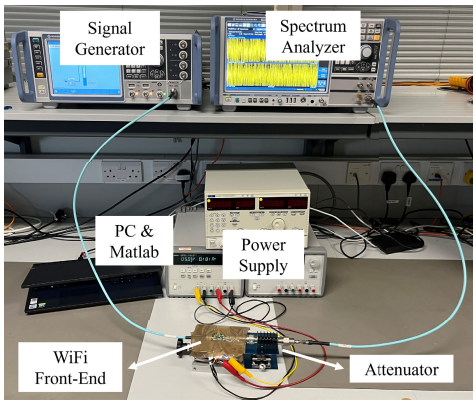


Fig. 8. Testbench setup.

per iteration step should be considered together to provide a more reasonable complexity comparison. When the enhanced Q-SPSA is applied on single-metric SSE optimization, that is, when $\alpha = 0.5$, the iteration step in (25) degenerates to

$$\mathbf{C}_{h+1} = \mathbf{C}_h - \frac{R\{\boldsymbol{\Delta}_h^H \mathbf{X}_h^H \mathbf{e}_h\}}{\boldsymbol{\Delta}_h^H \mathbf{X}_h^H \mathbf{X}_h \boldsymbol{\Delta}_h} \boldsymbol{\Delta}_h \tag{28}$$

and the complexity in terms of FLOPS for this step is

$$\beta_3 = N(4K + 8) + 4K - 1 \tag{29}$$

where detailed analysis can also be found in Appendix B. As it is compared with (25) which is for multimetric optimization, the extra complexity mainly comes from the FFT-related operations. Referring to (27) and (29), a ratio can be defined as the FLOPS consumed for each step of enhanced Q-SPSA for multimetric optimization over that for single-metric SSE optimization, that is,

$$\gamma = \frac{N(4K + 10\log_2 N - 8) + 4K + 23}{N(4K + 8) + 4K - 1} \tag{30}$$

where K is the number of DPD coefficients and N is the number of signal samples. An illustration of how the complexity ratio γ changes with different K and N values is given in Fig. 7. It is worth mentioning that we should pay attention to the parameter settings with larger K and N values as highlighted by the red box since in future systems, the PA behavior will become more complex due to wider bandwidth and therefore more complex models will be required. For small K and N values, the complexity for either single-metric or multimetric would be very low, which is of less necessity to analyze, and the LS algorithm could be favored without the need for iterative or multimetric methods.

In the zone of interest in Fig. 7, all the γ values are smaller than 1.35, for example, for $K = 100$ DPD coefficients and $N = 8192$ signal samples, γ is only approximately 1.28, which means that the extra cost for the multimetric step (25) when compared with the single-metric SSE step (28) is not that much and only 28%. In other words, to reach some requirement threshold, if the number of iterations for using multimetric is smaller than $1/\gamma$ of single-metric iterations, then the overall complexity of the multimetric would be lower, for example, to reach some multimetric requirement, if 100 iterations were used for multimetric method (25) and 300 iterations were used for single-metric method (28), considering $\gamma \approx 1.28$ for $K = 100$ and $N = 8192$, and then the overall complexity of using multimetric steps is only $100 \times 1.28/300 \approx 43\%$ of the overall complexity using single-metric method. The above method considers not only the number of iterations, but also FLOPS per iteration will also be used later for analyzing complexity in experimental results.

IV. Experimental Validation

A test bench based on a 5-GHz Wi-Fi nonlinear front-end module was used to validate the idea. The Wi-Fi front end was configured to the TX mode to linearize the PA inside. As shown in Fig. 8, an R&S vector signal generator (VSG) SMW200A was configured to generate different modulated signal sequences to feed into the front end. The output of the front end was attenuated and collected by an R&S spectrum analyzer FSW and sent back to the PC. The PC with MATLAB acted as a central controller to conduct automatic test routines for signal generation and data acquisition.

To emulate the online DPD identification scenario where the PA stimuli signal changes from time to time, different signal sequences must be generated. First, enough different 256-QAM orthogonal frequency division multiplexing (OFDM) signal sequences with sequence length $N = 8192$ were generated, with an over-sampling factor of 4, that is, the actual bandwidth of the signal would be 1/4 of the system sample rate deployed, for example, 40-MHz bandwidth for 160-MHz system sample rate, 160 MHz for 640-MHz sample rate. Then, to avoid performance fluctuation in online DPD identification scenario [20] in which PA input sequences are different, all of the sequences were processed using crest factor reduction (CFR) algorithm to make sure that they have similar peak-to-average power ratio (PAPR).

A. Test Case 1: 40-MHz Signal

A 40-MHz modulated signal with a 160-MHz system sample rate was first employed to validate the multimetric performance. The average input power fed into the front end was -15 dBm and the center frequency was 5.3 GHz.

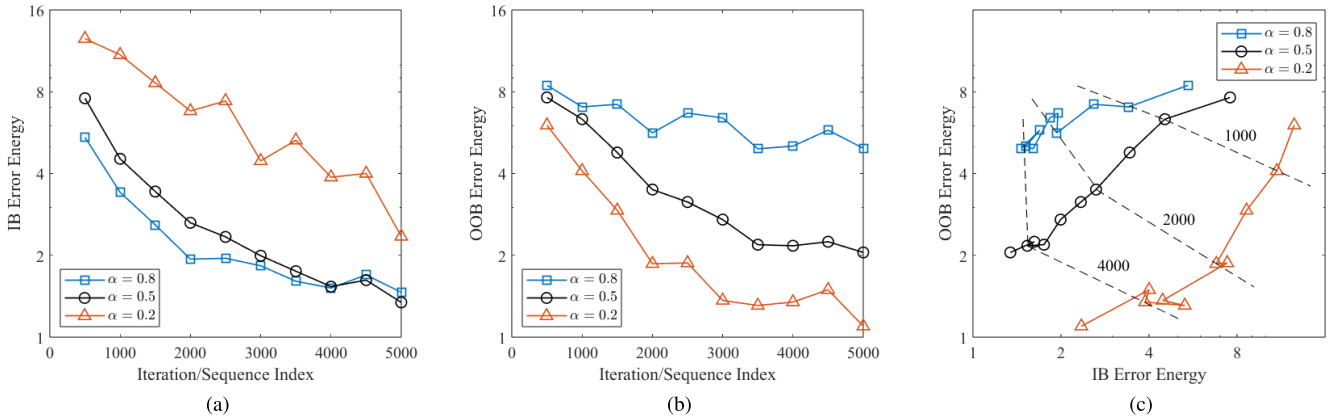


Fig. 9. Multimetric DPD performance. (a) IB error. (b) OOB error. (c) Performance trajectories.

Referring to (12), 3 settings of α with values $\{0.8, 0.5, 0.2\}$ were used. The larger the α, the more preference for the IB error, and vice versa. $\alpha = 0.5$ means equal consideration on IB and OOB error which corresponds to the conventional method [20] that only NMSE/SSE is used as the loss function, which is set as the benchmark for comparison. After that, for each setting of α, $H = 5000$ iterations were conducted, using 5000 different input sequences $\{\mathbf{x}_h\}$.

The linearity IB and OOB errors (between DPD input and PA output) are illustrated in Fig. 9(a) and (b), respectively. In Fig. 9(a), at the same number of iterations, it can be found that generally the larger the α (the more preference on IB), the lower IB error is acquired, meaning that the faster convergence on IB error. A similar phenomenon can be found in Fig. 9(b), the smaller the α (the more preference on OOB), also the faster convergence on OOB error. In Fig. 9(a), the IB error advantage of $\alpha = 0.8$ over $\alpha = 0.5$ become less and may reverse in later iterations (e.g., after 4000), which indicates that the iteration steps with $\alpha = 0.8$ and $\alpha = 0.5$ are approaching similar performance at convergence level of IB error (note that the y-axis is in log-scale), however, in the procedure, the $\alpha = 0.8$ iterations get significant IB error level advantage. To get a clear view of IB and OOB error simultaneously, the OOB versus IB error with the progression of iteration, namely, the multimetric performance trajectory, is provided in Fig. 9(c), in which each square, circle, or triangle marker marks the performance after every 500 iterations, that is, iterations 500, 1000, ..., 5000, and the dashed lines are equal iteration lines at iterations of 1000, 2000, and 4000. The iterations with three α settings will all descend the IB and OOB errors, but with different trends. At the same number of iterations, larger α tends to achieve lower IB error energy and higher OOB error energy, and vice versa.

Some details are also given in Table I, and it can be found that lower EVM that corresponds to IB error energy is achieved for larger α and lower adjacent channel power ratio (ACPR) that corresponds to OOB error energy is achieved for smaller α, at the same number of iterations. All of the above indicates that the proposed method can optimize IB and OOB linearity error with preference or priority by using different α. Comparison in some practical scenarios will also be given as follows.

TABLE I
MULTIMETRIC PERFORMANCE COMPARISON

Iter.	α	IB-Err.	EVM (%)	OOB-Err.	ACPR (dBc)
1k	0.8	3.41	2.05	7.03	-34.31/-37.61
	0.5*	4.53	2.37	6.34	-35.14/-38.12
	0.2	10.94	3.68	4.09	-37.21/-39.51
2k	0.8	1.94	1.56	5.64	-35.29/-37.73
	0.5*	2.63	1.83	3.49	-36.93/-40.03
	0.2	6.79	2.91	1.86	-40.26/-42.10
4k	0.8	1.52	1.41	5.05	-36.03/-37.69
	0.5*	1.53	1.41	2.17	-38.57/-41.98
	0.2	3.88	2.20	1.35	-41.29/-43.38

*Equivalent as the single-metric SSE optimization step in [20]

1) Scenario 1: Priority Optimization of Either Metric: As briefed in Section I, there could be different IB and OOB linearity requirements, and one of the practical scenarios would be either EVM or ACPR would be required to be optimized with priority. The following will provide the comparison of the only SSE targeted optimization (equivalent as $\alpha = 0.5$) as in [20], EVM priority optimization ($\alpha = 0.8$), and ACPR priority optimization ($\alpha = 0.2$). To provide a more reasonable complexity evaluation, the FLOPS consumed other than just the number of iterations are considered. In the test, the GMP model [16] with $K = 150$ coefficients is employed and each sequence is of length $N = 8192$. Substituting K and N value into (30), it can be found that $\gamma \approx 1.19$, which means that one multimetric step (25) ($\alpha = 0.8, 0.2$) takes 1.19 $\times$ of the complexity of one single-metric step ($\alpha = 0.5$) (28), where the multimetric step (25) with $\alpha = 0.5$ just degenerates to the simpler equivalent form in (28), namely, the single metric step, which is of less complexity.

Then, the EVM requirement is defined to be met when the EVM values on five consecutive sequences are all lower than the setting threshold. The complexity analysis for $\alpha = 0.8$ and $\alpha = 0.5$ (benchmark group) to satisfy some EVM requirements is given in Table II. In the table, the normalized complexity is the equivalent single-metric SSE optimization steps, for example, for multimetric steps with $\alpha \neq 0.5$, the normalized complexity is γ times the number of iterations used. The proposed method saves a considerable

TABLE II
EVM PRIORITY OPTIMIZATION

EVM without DPD (%) = 6.16				
EVM Requirement (%)	α	Iterations	Normalized Complexity	Reduction of FLOPS (%)
4.00	0.8 0.5*	32 89	38.08 89	57.21
3.00	0.8 0.5*	322 537	383.18 537	28.64
2.00	0.8 0.5*	1119 1677	1331.61 1677	20.60
1.50	0.8 0.5*	2395 3326	2850.05 3326	14.31

*Equivalent as the single-metric SSE optimization step in [20]

TABLE III
ACPR PRIORITY OPTIMIZATION

ACPR without DPD (dBc) = –32.26/-34.88				
ACPR Requirement (dBc)	α	Iterations	Normalized Complexity	Reduction of FLOPS (%)
-35.00	0.2 0.5*	328 1074	390.32 1074	63.66
-37.00	0.2 0.5*	1023 2221	1217.37 2221	45.19
-39.00	0.2 0.5*	1513 4382	1800.47 4382	58.91

*Equivalent as the single-metric SSE optimization step in [20]

number of iterations to reach the same EVM, and even if the extra complexity of the multimetric step is taken into account, the proposed method still saves appreciable FLOPS.

Similarly, ACPR requirements are defined to be met when both the lower and upper ACPR are lower than the threshold on five consecutive sequences. The comparison of $\alpha = 0.2$ and $\alpha = 0.5$ (benchmark group) is given in Table III. Not only the number of iterations, but also FLOPS are saved considerably, and the amount of saving is more significant than that on the IB error or EVM case in Table II.

From the EVM and ACPR priority optimization results presented in Tables II and III, it can be concluded that the proposed method with different α settings can reach either EVM or ACPR requirements with significantly lower complexity, in terms of not only the number of iterations, but also FLOPS consumption.

2) Scenario 2: Meeting Arbitrary Multimetric Requirements: Real systems may impose requirements on multiple metrics simultaneously. Let us assume that there are 36 combinations of multimetric requirements in terms of EVM and ACPR, with EVM requirements from 1.50% to 4.00% with the step of 0.5% and ACPR requirements (both upper and lower) from -33 to -38 dBc with the step of 1 dBc. Similarly, the requirement is defined to be met when five consecutive sequences all satisfy it, and normalized complexity considering FLOPS is considered. Fig. 10 reveals the best α settings to reach arbitrary multimetric requirement combinations at different coordinates using the minimum normalized complexity.

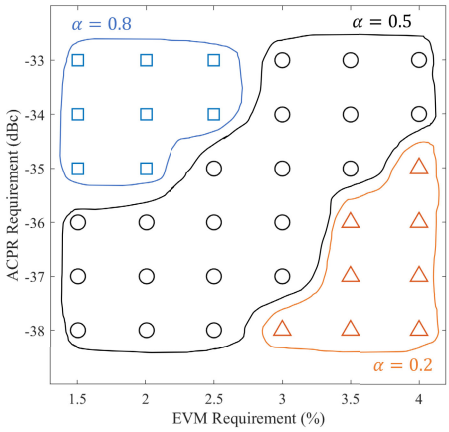


Fig. 10. α settings reaching different multimetric requirements with minimum normalized complexity.

It can be found that the iterations with $\alpha = 0.8$ (higher IB preference) are good at the multimetric requirements with stricter IB or EVM, and vice versa for the ones with $\alpha = 0.2$ (higher OOB preference), while the ones with $\alpha = 0.5$ are better at relatively balanced EVM and ACPR requirements. This indicates that for unbalanced multimetric requirements, the conventional SSE-only considered method [20] will consume more FLOPS and be unfavorable, and the proposed method with different α settings can reach these unbalanced requirements with less complexity.

It is worth mentioning here that the specific settings of α to reach some given multimetric for EVM and ACPR would be a very practical problem. A feedback control mechanism may be employed. Specifically, performance check can be conducted routinely to evaluate the current distance to the given multimetric requirement, and α can be adjusted accordingly in which higher weighting can be applied to the target that is still "far away" while lower weighting to the one that is nearly met.

3) Scenario 3: SSE-Monitored DPD System: Indicators such as EVM, which is an end-to-end quality metric, and the linearization error in some bands, may not be easily accessible to the DPD system, and instead, the DPD system may only monitor the energy of the linearity error vector $\mathbf{e}$ by using an indicator such as SSE (2) or NMSE. Thus, the following will provide a comparison when the optimizations with different weightings in different frequency regions reach the same SSE. As shown in Table IV, at almost the same SSE level, $\alpha = 0.8$ gets lower IB error and EVM, and higher OOB error and ACPR, while $\alpha = 0.2$ gets lower OOB error and ACPR, and higher IB error and EVM. The error power spectrum density (PSD) plots at SSE threshold 5.00 for the three α settings is given in Fig. 11, which indicates that at almost the same SSE/NMSE level, the frequency distribution of error could be totally different. The proposed method with $\alpha = 0.8$ can find the DPD coefficients with lower IB error, and the method with $\alpha = 0.2$ can find the ones with lower OOB error. Fig. 12 shows the IB error advantage, or EVM advantage for the proposed method with $\alpha = 0.8$ over the benchmark group with $\alpha = 0.5$ that corresponds to [20]. Fig. 13 demonstrates the OOB advantage, or ACPR advantage

TABLE IV
SAME SSE COMPARISON

SSE without DPD = 47.76						
SSE Threshold	SSE	α	IB Error	EVM (%)	OOB Error	ACPR (dBc)
10.00	9.00	0.8	3.26	1.97	5.73	-35.18/-37.95
	9.92	0.5*	4.48	2.42	5.44	-35.39/-37.65
	9.70	0.2	7.34	2.96	2.36	-39.81/-41.29
5.00	4.99	0.8	1.25	1.28	3.74	-37.48/-38.59
	4.84	0.5*	1.96	1.57	2.88	-37.90/-40.22
	4.74	0.2	3.63	2.10	1.11	-42.36/-43.83

*Equivalent as the single-metric SSE optimization step in [20]

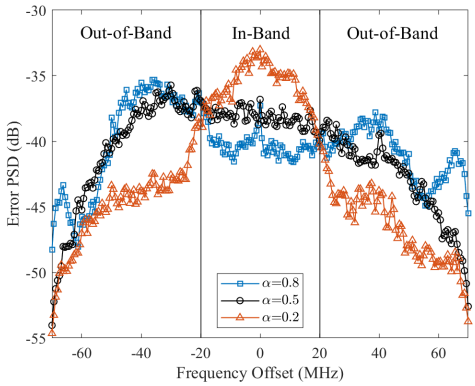


Fig. 11. Error PSDs at SSE threshold 5.00.

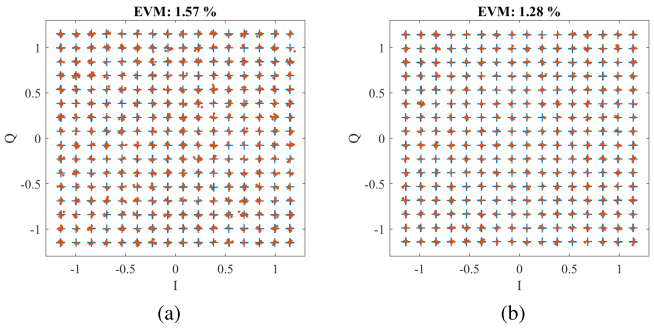


Fig. 12. IB/EVM advantage at SSE threshold 5.00. (a) $\alpha = 0.5$ and EVM = 1.57%. (b) $\alpha = 0.8$ and EVM = 1.28%.

for the proposed method with $\alpha = 0.2$ over the benchmark with $\alpha = 0.5$ and also the case with $\alpha = 0.8$.

In addition, to demonstrate that the proposed method can also be applied to scenarios with multiple frequency divisions that are more complicated than IB/OOB, a supplemental experiment was also conducted. As depicted in Fig. 14, the proposed method applies $\alpha_1 = 0.8$ weighting in multiple frequency divisions designated by the solid double arrows, and applies $\alpha_2 = 0.2$ elsewhere, while and the benchmark method with $\alpha = 0.5$ [20] means equal weighting everywhere as a comparison. It can be found that the proposed method can also distinguish the error in multiple frequency divisions, not limited to IB/OOB.

B. Test Case 2: EVM Priority Optimization of the 160-MHz Signal

To validate the performance of the proposed method for wider bandwidth signals, another test was conducted employing a 160-MHz signal with a 640-MHz system sample rate at the center frequency of 5.57 GHz (Wi-Fi channel 114)

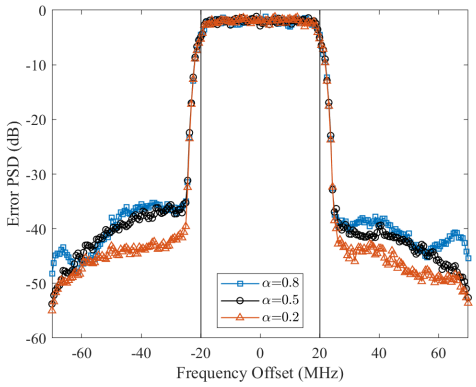


Fig. 13. OOB/ACPR advantage at SSE threshold 5.00.

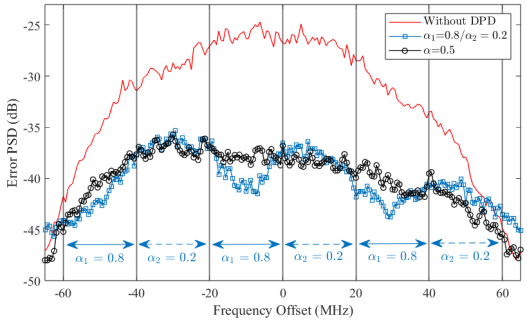


Fig. 14. Error PSDs at SSE threshold 5.00 for multiple frequency divisions.

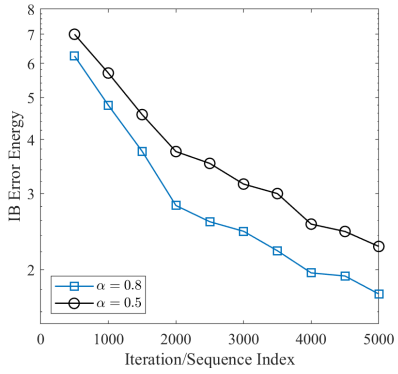


Fig. 15. IB error comparison.

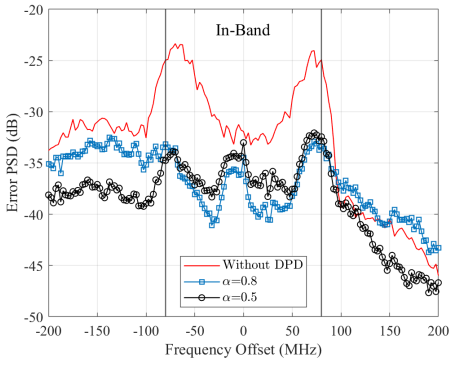


Fig. 16. Error PSDs at iteration 2000.

TABLE V
EVM PRIORITY OPTIMIZATION

EVM without DPD (%) = 4.85				
EVM Requirement (%)	α	Iterations	Normalized Complexity	Reduction of FLOPS (%)
2.50	0.8 0.5*	600 978	714.00 978	26.99
2.00	0.8 0.5*	1452 2145	1727.88 2145	19.45
1.75	0.8 0.5*	2109 3604	2509.71 3604	30.36

*Equivalent as the single-metric SSE optimization step in [20]

and input power of −12 dBm. Considering the upper and lower ACPR (480-MHz range) for a 160-MHz signal is very unbalanced, which will make the situation much more complicated, thus, only EVM priority optimization-related results will be given. The IB error energy comparison is given in Fig. 15. When compared with the 40-MHz test case shown in Fig. 9(a), in the "tougher" 160-MHz case, the proposed method with $\alpha = 0.8$ reveals a greater advantage in IB error energy optimization. The error PSD comparison at iteration 2000 is also provided in Fig. 16, and it can be found that in the IB region (from −80- to +80-MHz offset), the error PSD of $\alpha = 0.8$ is lower. Similar to Table II, normalized complexity comparison to reach some EVMs is given in Table V.

V. CONCLUSION

In this article, a multimetric DPD concept is proposed which enables that linearity performance in different frequency regions can be approached with desired preference. This is especially beneficial in terms of approaching speed and overall complexity in cases where unbalanced linearity requirements in different frequency regions are needed. The method is implemented in an online DPD identification scenario in which the input signal sequences for training DPD coefficients change from time to time. A mathematical form to evaluate the linearity error in different frequency regions is first established and then the Q-SPSA is employed as an efficient optimization algorithm and derived for online multimetric DPD identification, by distinguishing and weighting the error energy in different frequency regions. Meanwhile, an enhanced calculation form of the iteration step is also derived, which is mathematical equivalent but of less complexity and better numerical stability.

Experimental results of 40- and 160-MHz modulated signals running on a 5-GHz Wi-Fi nonlinear front-end validate that the proposed method can realize priority optimization of different metrics. Linearity requirements in different frequency regions can be met with considerably less complexity in terms of not only the number of iterations, but also FLOPS. Also, at the same SSE/NMSE level, better linearity performance in desired frequency regions can be achieved by using the proposed approach, when compared with the conventional method that considers SSE/NMSE only in the optimization. As a proof of concept, this work only employs fixed weightings on error energy in different frequency divisions. While adaptive weightings can be more favorable for some specific multimetric requirements given, the control mechanism of weightings would become more complex, especially for three or more metric requirements. We leave that in future work.

APPENDIX A
DERIVATION OF ENHANCED Q-SPSA

The three loss function measurements (23) can be expressed in fully expansion form as follows:

$$\begin{aligned} L(+\boldsymbol{\Delta}_h) &= \mathbf{e}_h^H\mathbf{F}^H\boldsymbol{\Psi}\mathbf{F}\mathbf{e}_h + \boldsymbol{\Delta}_h^H\mathbf{X}_h^H\mathbf{F}^H\boldsymbol{\Psi}\mathbf{F}\mathbf{e}_h \\ &\quad + \mathbf{e}_h^H\mathbf{F}^H\boldsymbol{\Psi}\mathbf{F}\mathbf{X}_h\boldsymbol{\Delta}_h + \boldsymbol{\Delta}_h^H\mathbf{X}_h^H\mathbf{F}^H\boldsymbol{\Psi}\mathbf{F}\mathbf{X}_h\boldsymbol{\Delta}_h \\ L(-\boldsymbol{\Delta}_h) &= \mathbf{e}_h^H\mathbf{F}^H\boldsymbol{\Psi}\mathbf{F}\mathbf{e}_h - \boldsymbol{\Delta}_h^H\mathbf{X}_h^H\mathbf{F}^H\boldsymbol{\Psi}\mathbf{F}\mathbf{e}_h \\ &\quad - \mathbf{e}_h^H\mathbf{F}^H\boldsymbol{\Psi}\mathbf{F}\mathbf{X}_h\boldsymbol{\Delta}_h + \boldsymbol{\Delta}_h^H\mathbf{X}_h^H\mathbf{F}^H\boldsymbol{\Psi}\mathbf{F}\mathbf{X}_h\boldsymbol{\Delta}_h \\ L(\mathbf{0}) &= \mathbf{e}_h^H\mathbf{F}^H\boldsymbol{\Psi}\mathbf{F}\mathbf{e}_h. \end{aligned} \tag{A.1}$$

Then the numerator and the denominator in (22) can be derived as follows:

$$\begin{aligned} L(+\boldsymbol{\Delta}_h) - L(-\boldsymbol{\Delta}_h) &= 2\big(\boldsymbol{\Delta}_h^H\mathbf{X}_h^H\mathbf{F}^H\boldsymbol{\Psi}\mathbf{F}\mathbf{e}_h \\ &\quad + \mathbf{e}_h^H\mathbf{F}^H\boldsymbol{\Psi}\mathbf{F}\mathbf{X}_h\boldsymbol{\Delta}_h\big) \qquad \text{(A.2)} \\ &= 4R\big\{\boldsymbol{\Delta}_h^H\mathbf{X}_h^H\mathbf{F}^H\boldsymbol{\Psi}\mathbf{F}\mathbf{e}_h\big\} \qquad \text{(A.3)} \end{aligned}$$

and

$$L(+\boldsymbol{\Delta}_h) - 2L(\mathbf{0}) + L(-\boldsymbol{\Delta}_h) = 2\boldsymbol{\Delta}_h^H\mathbf{X}_h^H\mathbf{F}^H\boldsymbol{\Psi}\mathbf{F}\mathbf{X}_h\boldsymbol{\Delta}_h \tag{A.4}$$

respectively, where $R\{\cdot\}$ is the real part operator. Derivation from (A.2) to (A.3) can be made since $\boldsymbol{\Delta}_h^H\mathbf{X}_h^H\mathbf{F}^H\boldsymbol{\Psi}\mathbf{F}\mathbf{e}_h$ and $\mathbf{e}_h^H\mathbf{F}^H\boldsymbol{\Psi}\mathbf{F}\mathbf{X}_h\boldsymbol{\Delta}_h$ are both scalar complex numbers and they conjugate. Finally, substituting (A.3) and (A.4) into (22) and (24), the enhanced Q-SPSA progression step can be derived as follows:

$$\mathbf{C}_{h+1} = \mathbf{C}_h - \frac{R\big\{\boldsymbol{\Delta}_h^H\mathbf{X}_h^H\mathbf{F}^H\boldsymbol{\Psi}\mathbf{F}\mathbf{e}_h\big\}}{\boldsymbol{\Delta}_h^H\mathbf{X}_h^H\mathbf{F}^H\boldsymbol{\Psi}\mathbf{F}\mathbf{X}_h\boldsymbol{\Delta}_h}\boldsymbol{\Delta}_h. \tag{A.5}$$

APPENDIX B
DETAILED COMPLEXITY ANALYSIS

First of all, the conversion between different operations and the number of FLOPS consumed is defined as follows.

1) *Complex multiplication:* Six FLOPS.
2) *Complex addition/subtraction:* Two FLOPS.
3) *Real multiplication/division:* One FLOPS.
4) *Real addition/subtraction:* One FLOPS.

A. Conventional Step for Multimetric Optimization

The conventional iteration step for multimetric optimization (24) with (22) is

$$\mathbf{C}_{h+1} = \mathbf{C}_h - \frac{L(+\boldsymbol{\Delta}_h) - L(-\boldsymbol{\Delta}_h)}{2\big(L(+\boldsymbol{\Delta}_h) - 2L(\mathbf{0}) + L(-\boldsymbol{\Delta}_h)\big)}\boldsymbol{\Delta}_h \tag{B.1}$$

and corresponding complexity analysis step-by-step will be given as follows.

Step 1.1: Compute $\mathbf{e}_h = \mathbf{y}_h - \mathbf{x}_h$*:* $\mathbf{e}_h$, $\mathbf{x}_h$, and $\mathbf{y}_h$ are all of size $N \times 1$, where N is the number of signal samples. N

complex additions are conducted, and $2K$ FLOPS is for this step.

Step 1.2: Compute $\mathbf{X}_h\boldsymbol{\Delta}_h$*:* $\mathbf{X}_h$ is of size $N \times K$, where K is number of DPD coefficients and $\boldsymbol{\Delta}_h$ is of size $K \times 1$. This step should not be regarded as a complex $N \times K$ matrix multiplies a complex $K \times 1$ vector, which uses NK complex multiplications and $N(K-1)$ complex additions. Considering each element of $\boldsymbol{\Delta}_h$ vector must be one of $\{1+1j, 1-1j, -1+1j, -1-1j\}$ [19], [20] which are actually adding/subtracting instead of multiplying elements in $\mathbf{X}_h$, and

$$\mathbf{X}_h\boldsymbol{\Delta}_h = R\{\mathbf{X}_h\}R\{\boldsymbol{\Delta}_h\} - I\{\mathbf{X}_h\}I\{\boldsymbol{\Delta}_h\} + j(R\{\mathbf{X}_h\}I\{\boldsymbol{\Delta}_h\} + I\{\mathbf{X}_h\}R\{\boldsymbol{\Delta}_h\}) \quad \text{(B.2)}$$

where $R\{\cdot\}$ and $I\{\cdot\}$ are the real and imagine part operator. $4N(K-1)+2N = N(4K-2)$ FLOPS is for this step.

Step 1.3: Compute $\mathbf{e}_h^+ = \mathbf{e}_h + \mathbf{X}_h\boldsymbol{\Delta}_h$ *and* $\mathbf{e}_h^- = \mathbf{e}_h - \mathbf{X}_h\boldsymbol{\Delta}_h$*:* $2N$ complex additions involved, and $4N$ FLOPS is for this step.

Step 1.4: Compute $\mathbf{Fe}_h$*,* $\mathbf{Fe}_h^+$*, and* $\mathbf{Fe}_h^-$*:* As for the product of an $N \times N$ DFT matrix and an $N \times 1$ vector, counting it as product of complex $N \times N$ matrix and complex $N \times 1$ vector will overestimate the FLOPS. Instead, thanks to the fast Fourier transform (FFT) algorithm, the DFT can be calculated with only $N\log_2 N$ complex additions and $N(\log_2 N-3)/2+2$ complex multiplications [28]. Then the FLOPS for this step is $N(15\log_2 N - 27) + 36$.

Step 1.5: Compute $L = (\mathbf{Fe}_h)^H\boldsymbol{\Psi}(\mathbf{Fe}_h)$*,* $L^+ = (\mathbf{Fe}_h^+)^H\boldsymbol{\Psi}(\mathbf{Fe}_h^+)$*, and* $L^- = (\mathbf{Fe}_h^-)^H\boldsymbol{\Psi}(\mathbf{Fe}_h^-)$*:* In this step, we are sure that the product must be a real number. To count operations conducted, let us first analyze the calculation of the real part of the product of two complex numbers, that is,

$$R\{(a_1 + jb_1)(a_2 + jb_2)\} = a_1a_2 - b_1b_2 \quad \text{(B.3)}$$

from which it can be found that only two real multiplications and one real addition are conducted per product. Thus, for the real part of the product of a complex vector of $1 \times N$, real diagonal matrix of $N \times N$, and a complex vector of $N \times 1$, $2N$ real multiplications and N real additions will be conducted for getting the real parts, and another N real multiplications will be conducted for multiplying the diagonal entries, and finally $N-1$ real additions will be conducted for summing them up, where $3N$ real multiplications and $2N-1$ real additions are conducted. In total, $15N-3$ FLOPS is for this step.

Step 1.6: Compute $A = L^+ - L^-$ *and* $B = L^+ - 2L + L^-$*:* Four FLOPS is for this step.

Step 1.7: Compute $\mathbf{C}_{h+1} = \mathbf{C}_h - A/(2B) * \boldsymbol{\Delta}_h$*:* $2K+2$ real multiplication is involved for getting $A/(2B) * \boldsymbol{\Delta}_h$ and N complex additions involved later. $4K+2$ FLOPS is for this step.

Overall: $N(4K + 15\log_2 N - 8) + 4K + 39$ FLOPS is involved.

B. Enhanced Step for Multimetric Optimization

The proposed enhanced iteration step for multimetric optimization (25) is

$$\mathbf{C}_{h+1} = \mathbf{C}_h - \frac{R\{\boldsymbol{\Delta}_h^H\mathbf{X}_h^H\mathbf{F}^H\boldsymbol{\Psi}\mathbf{Fe}_h\}}{\boldsymbol{\Delta}_h^H\mathbf{X}_h^H\mathbf{F}^H\boldsymbol{\Psi}\mathbf{F}\mathbf{X}_h\boldsymbol{\Delta}_h}\boldsymbol{\Delta}_h \quad \text{(B.4)}$$

and corresponding complexity analysis will be given as follows.

Step 2.1: Compute $\mathbf{e}_h = \mathbf{y}_h - \mathbf{x}_h$*:* $2N$ FLOPS is for this step.

Step 2.2: Compute $\mathbf{X}_h\boldsymbol{\Delta}_h$*:* Similar to *Step 1.2*, $N(4K-2)$ FLOPS is for this step.

Step 2.3: Compute $\mathbf{v}_1 = \mathbf{Fe}_h$*:* Similar to *Step 1.4*, $N(5\log_2 N - 9) + 12$ FLOPS is for this step.

Step 2.4: Compute $\mathbf{v}_2 = \mathbf{F}(\mathbf{X}_h\boldsymbol{\Delta}_h)$*:* $N(5\log_2 N - 9) + 12$ FLOPS is for this step.

Step 2.5: Compute $A = R\{\mathbf{v}_1^H\boldsymbol{\Psi}\mathbf{v}_2\}$*:* Similar to *Step 1.5*, we are only interested in the real part of the product. $5N-1$ FLOPS is for this step.

Step 2.6: Compute $B = \mathbf{v}_1^H\boldsymbol{\Psi}\mathbf{v}_1$*:* $5N-1$ FLOPS is for this step.

Step 2.7: Compute $\mathbf{C}_{h+1} = \mathbf{C}_h - A/B * \boldsymbol{\Delta}_h$*:* Similar to *Step 1.7*, $4K+1$ FLOPS is for this step.

Overall: $N(4K + 10\log_2 N - 8) + 4K + 23$ FLOPS is involved.

C. Enhanced Step for Single-Metric SSE Optimization

The proposed enhanced iteration step for single-metric optimization (28) is

$$\mathbf{C}_{h+1} = \mathbf{C}_h - \frac{R\{\boldsymbol{\Delta}_h^H\mathbf{X}_h^H\mathbf{e}_h\}}{\boldsymbol{\Delta}_h^H\mathbf{X}_h^H\mathbf{X}_h\boldsymbol{\Delta}_h}\boldsymbol{\Delta}_h \quad \text{(B.5)}$$

and corresponding complexity analysis will be given as follows.

Step 3.1: Compute $\mathbf{e}_h = \mathbf{y}_h - \mathbf{x}_h$*:* $2N$ FLOPS is for this step.

Step 3.2: Compute $\mathbf{X}_h\boldsymbol{\Delta}_h$*:* Similar to *Step 1.2*, $N(4K-2)$ FLOPS is for this step.

Step 3.3: Compute $A = R\{(\mathbf{X}_h\boldsymbol{\Delta}_h)^H\mathbf{e}_h\}$*:* Similar to *Step 2.5* and *Step 2.6*, and note that no $\boldsymbol{\Psi}$ is involved here, so $4N-1$ FLOPS is for this step.

Step 3.4: Compute $B = (\mathbf{X}_h\boldsymbol{\Delta}_h)^H(\mathbf{X}_h\boldsymbol{\Delta}_h)$*:* Similar to *Step 3.3*, $4N-1$ FLOPS is for this step.

Step 3.5: Compute $\mathbf{C}_{h+1} = \mathbf{C}_h - A/B * \boldsymbol{\Delta}_h$*:* $4K+1$ FLOPS is for this step.

Overall: $N(4K+8) + 4K - 1$ FLOPS is involved.

REFERENCES

[1] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May 2020.

[2] C. Deng et al., "IEEE 802.11be Wi-Fi 7: New challenges and opportunities," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2136–2166, 4th Quart., 2020.

[3] *IEEE 802.11be Technology Introduction*, Rohde & Schwarz, Munich, Germany, 2022.

[4] E. Khorov, I. Levitsky, and I. F. Akyildiz, "Current status and directions of IEEE 802.11be, the future Wi-Fi 7," *IEEE Access*, vol. 8, pp. 88664–88688, 2020.

[5] 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.

[6] D. F. Kimball et al., "High-efficiency envelope-tracking W-CDMA base-station amplifier using GaN HFETs," *IEEE Trans. Microw. Theory Techn.*, vol. 54, no. 11, pp. 3848–3856, Nov. 2006.

[7] O. Hammi and F. M. Ghannouchi, "Twin nonlinear two-box models for power amplifiers and transmitters exhibiting memory effects with application to digital predistortion," *IEEE Microw. Wireless Compon. Lett.*, vol. 19, no. 8, pp. 530–532, Aug. 2009.

[8] K. Freiberger, M. Wolkerstorfer, H. Enzinger, and C. Vogel, "Digital predistorter identification based on constrained multi-objective optimization of WLAN standard performance metrics," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2015, pp. 862–865.

[9] Y. Li et al., "A 15-bit quadrature digital power amplifier with transformer-based complex-domain efficiency enhancement," *IEEE J. Solid-State Circuits*, vol. 57, no. 6, pp. 1610–1622, Jun. 2022.

[10] S. Wan et al., "A high-efficiency two-stage GaAs HBT Doherty power amplifier with thermal compensation for WLAN application," in *IEEE MTT-S Int. Microw. Symp. Dig.*, May 2021, pp. 1–3.

[11] A. Garcia-Rodriguez, D. López-Pérez, L. Galati-Giordano, and G. Geraci, "IEEE 802.11be: Wi-Fi 7 strikes back," *IEEE Commun. Mag.*, vol. 59, no. 4, pp. 102–108, Apr. 2021.

[12] M. Yang, B. Li, Z. Yan, and Y. Yan, "AP coordination and full-duplex enabled multi-band operation for the next generation WLAN: IEEE 802.11be (EHT)," in *Proc. 11th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2019, pp. 1–7.

[13] Y. Liu, Y. Yu, Z. Du, and L. Cuthbert, "Sequential state Q-learning uplink resource allocation in multi-AP 802.11be network," in *Proc. IEEE 96th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2022, pp. 1–5.

[14] P. Imputato, S. Avallone, and D. Magrin, "Multi-AP coordination in Wi-Fi 7 exploiting time resources sharing," in *Proc. IEEE Int. Medit. Conf. Commun. Netw. (MeditCom)*, Sep. 2022, pp. 166–171.

[15] 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.

[16] D. R. Morgan, Z. Ma, J. Kim, M. G. Zierdt, and J. Pastalan, "A generalized memory polynomial model for digital predistortion of RF power amplifiers," *IEEE Trans. Signal Process.*, vol. 54, no. 10, pp. 3852–3860, Oct. 2006.

[17] A. Zhu, J. C. Pedro, and T. J. Brazil, "Dynamic deviation reduction-based Volterra behavioral modeling of RF power amplifiers," *IEEE Trans. Microw. Theory Techn.*, vol. 54, no. 12, pp. 4323–4332, Dec. 2006.

[18] 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.

[19] 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.

[20] 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.

[21] J. Zhai et al., "The threshold optimization of the canonical piecewise linear function-based model with a modified quadratic SPSA," *IEEE Microw. Wireless Compon. Lett.*, vol. 31, no. 6, pp. 612–615, Jun. 2021.

[22] C. Yu, L. Guan, E. Zhu, and A. Zhu, "Band-limited Volterra series-based digital predistortion for wideband RF power amplifiers," *IEEE Trans. Microw. Theory Techn.*, vol. 60, no. 12, pp. 4198–4208, Dec. 2012.

[23] X. Xia, X. Quan, Y. Liu, S. Shao, and Y. Tang, "A frequency-selective digital predistortion method based on a generalized indirect learning architecture," *IEEE Trans. Signal Process.*, vol. 70, pp. 2334–2348, 2022.

[24] H. Yin, C. Chu, and A. Zhu, "Accelerating model adaptation of multi-metric digital predistortion for RF power amplifiers using composited quadratic loss function," in *Proc. 52nd Eur. Microw. Conf. (EuMC)*, Sep. 2022, pp. 345–348.

[25] 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.

[26] L. Ding, F. Mujica, and Z. Yang, "Digital predistortion using direct learning with reduced bandwidth feedback," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Jun. 2013, pp. 1–3.

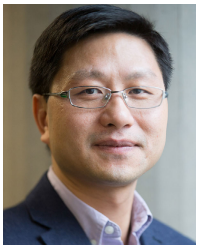
[27] D. Goldberg, "What every computer scientist should know about floating-point arithmetic," *ACM Comput. Surveys*, vol. 23, no. 1, pp. 5–48, Mar. 1991.

[28] R. Yavne, "An economical method for calculating the discrete Fourier transform," in *Proc. AFIPS Fall Joint Comput. Conf.*, Dec. 1968, pp. 115–125.

Hang Yin (Graduate Student Member, IEEE) was born in Shenyang, China, in 1999. He received the B.E. degree in information engineering and the M.E. degree in electromagnetic fields and microwave technology from Southeast University (SEU), Nanjing, China, in 2018 and 2021, respectively. He is currently pursuing the Ph.D. degree with University College Dublin (UCD), Dublin, Ireland.

His research interests include behavioral modeling and linearization of radio-frequency (RF) power amplifiers (PAs) and multiple-input–multiple-output (MIMO) transmitters, adaptive on-demand low-complexity digital predistortion (DPD) algorithms for future complex scenarios, and software-defined radio (SDR).

Mr. Yin was a recipient of the IEEE Microwave Theory and Technology Society (MTT-S) Graduate Fellowship in 2023, and the Rohde and Schwarz Graduate Fellowship in 2020.



Anding Zhu (Fellow, IEEE) received the Ph.D. degree in electronic engineering from the 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.

Prof. Zhu is an Elected Member of the IEEE Microwave Theory and Technology Society (MTT-S) Administrative Committee (AdCom) and a Member of the IEEE Future Directions Committee. He served as the Secretary of the IEEE MTT-S AdCom in 2018 and the Chair of the IEEE MTT-S Microwave High-Power Techniques Committee (TC-12) in 2020 and 2021. He was a recipient of the 2021 IEEE MTT-S Microwave Prize. He served as a Track Editor for IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES in 2020–2022.