Boosted Model Tree-Based Behavioral Modeling for Digital Predistortion of RF Power Amplifiers

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

***Abstract*— In this article, we propose a new behavioral modeling approach, called boosted model tree, to characterize and compensate for the complex nonlinear distortions induced by wideband high-efficiency radio frequency power amplifiers. With the proposed model, the input data are classified into different zones by decision trees and each zone is assigned separate submodels. We also employ a model boosting technique to build multiple parallel tree structures that jointly model the desired nonlinear behavior. By designing dedicated optimization procedures, both tree structures and submodel coefficients can be efficiently identified. It is demonstrated that the combination of piecewise and parallel structures provides a powerful and hardware-efficient way to model nonlinear memory effect and cross terms. Based on the experimental results, the proposed method can achieve improved linearization performance with low hardware complexity under challenging wideband predistortion scenarios.**

***Index Terms*— Behavioral modeling, boosting, decision tree, digital predistortion (DPD), machine learning, power amplifier (PA).**

I. Introduction

IN THE 5G era, new technologies are expected to be deployed in mobile communication systems to satisfy the growing demand for better wireless networks. Key technologies, including new waveforms, advanced multiple-input–multiple-output (MIMO), increased bandwidth, and network densification, have been developed to meet the diverse requirements of the next-generation network [1]. Implementing these new features while keeping high energy efficiency thus becomes a challenging yet important task in the design of future wireless communication systems.

Digital predistortion (DPD) is a widely adopted linearization technique for mitigating nonlinearities caused by radio frequency (RF) power amplifiers (PAs) [2], [3]. It allows PAs to be operated at higher power levels for higher efficiency without losing linearity, enabling a more efficient wireless system. Nevertheless, the new specifications and operation principles of 5G transmitters pose challenges to the design of DPD [4]. On one hand, wider bandwidth and new high-efficiency PA architectures lead to more complex PA characteristics [5], which requires more sophisticated DPD models for linearization. On the other hand, the use of MIMO and network densification [6] techniques reduces the output power of individual PAs, which lowers the power budget of DPDs. To tackle these challenges, DPD systems with improved performance and lower power consumption are desired.

In the past two decades, many DPD models have been developed. Many classic models, e.g., memory polynomials (MPs) [7], generalized MP (GMP) [8], and dynamic deviation reduction (DDR) [9], are based on the Volterra series. Such polynomial-based models provide an intuitive description of the PA nonlinearity, which eases the development of interpretable models. The use of polynomials as basis functions, however, suffers from high hardware complexity in DPD implementation and potential ill-conditioning in model extraction. Under this context, another important modeling technique, lookup table (LUT), has regained popularity due to its low hardware complexity [10]–[12]. The spacing of LUTs can also be optimized to have a nonuniform pattern for better accuracy [13], [14], but the achieved performance is still limited by its simple structure. Other basis functions have also been developed, e.g., decomposed vector rotation [15] and splines [16], leading to new models with improved performance and reduced complexity.

Besides the search for better nonlinear basis functions, a few techniques have been proposed to boost performance by aggregating a collection of different models. Parallel modeling techniques include basis functions from multiple models to generate a larger and more accurate behavioral model [17], [18]. A different method is to arrange different models in cascade [19], where the output of one model becomes the input to the next. Another approach, piecewise modeling method, keeps a number of models and selects one model each time based on the input signal characteristics [20]–[22].

In this article, a novel behavioral modeling technique, called boosted model tree (BMT), is developed to enhance the linearization performance of conventional linear-in-parameters DPD models. To achieve improved accuracy and lower complexity, we use small-size pruned Volterra model as the basic element and aggregate many of them using data-driven machine learning techniques. Specifically, we design a piecewise model structure, named model tree, by using decision trees to select the suitable submodel for each input data sample. To further improve the performance, we employ a model

Manuscript received March 28, 2021; revised April 27, 2021; accepted April 28, 2021. Date of publication May 27, 2021; date of current version September 2, 2021. This work was supported in part by the Science Foundation Ireland under Grant 16/IA/4449, Grant 17/NSFC/4850, and Grant 13/RC/2077_P2. *(Corresponding author: Yue Li.)*

Yue Li, Xiaoyu Wang, and Anding Zhu 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).

Jingzhou Pang is with the School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China (e-mail: jingzhou.pang@cqu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMTT.2021.3081096.

Digital Object Identifier 10.1109/TMTT.2021.3081096

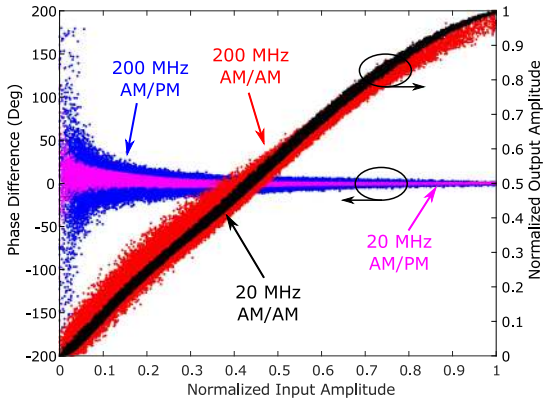

Fig. 1. AM-AM and AM-PM characteristics using 20- and 200-MHz signals.

boosting technique to build multiple model trees with distinct tree structures and use them in parallel to produce the output signal. The final BMT model with both tree-based piecewise structure and boosting-based parallel architecture is shown to provide strong linearization performance while consuming few hardware resources. Dedicated model training algorithms are also proposed to train both tree structures and submodel coefficients efficiently. Compared with the exhaustive search strategy, the proposed optimization techniques can achieve satisfactory linearization performance with significantly lower complexity.

The rest of this article is organized as follows. The background of nonlinear memory effect modeling is briefly discussed in Section II. Section III describes the proposed model structure and explains the use of the two machine learning techniques, i.e., decision tree and model boosting. Subsequently, in Section IV, we show the training method for the proposed model. The experimental results and complexity analysis are reported in Section V, followed by a conclusion in Section VI.

II. Background

The evolution of emerging RF systems and communication standards has created significant new challenges for DPD model development. Generally speaking, the reasons are twofold. The first factor is the widespread adoption of complex PA architectures, such as multiband/multimode and multistage Doherty [23]–[25] and load-modulated balanced amplifiers (LMBAs) [26]–[28]. While these techniques have greatly improved the energy efficiency of modern RF systems, these high-efficiency PAs are more difficult to model, because of the sophisticated interaction between their internal blocks. In particular, they often exhibit very different nonlinear characteristics when driven to different power ranges.

The second consideration is the wider signal bandwidth, which requires the DPD model to guarantee accurate frequency response over a wider frequency range. Moreover, due to the bandwidth limitation in circuit components, PAs typically exhibit more complicated characteristics with wideband excitations. As discussed in [29], such frequency-dependent nonlinearity can strongly affect the linearizability of PAs. As shown in Fig. 1, with the same PA, the AM-AM and AM-PM characteristics of 200-MHz signals have significantly more dispersions than that with 20-MHz signals.

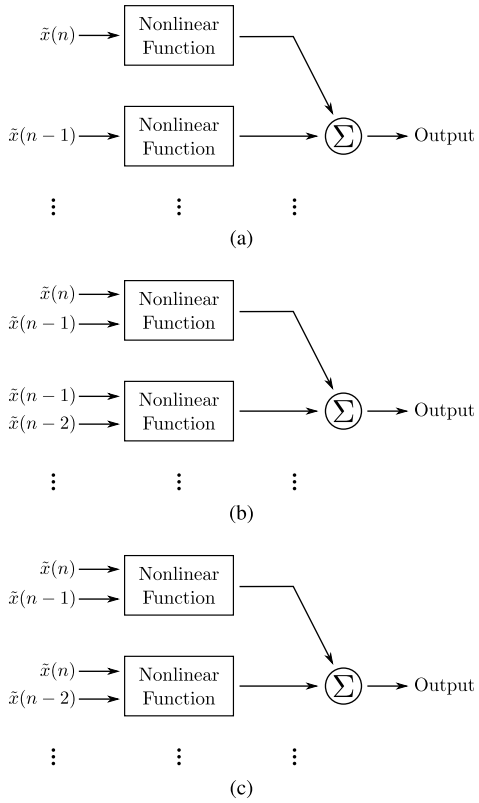

Fig. 2. Different memory effect representation in existing DPD models: (a) MP, (b) GMP, and (c) DDR.

To tackle these issues, DPD models should be designed to have more powerful capability to represent PA's memory effects. Therefore, the design of DPD models must consider more complicated nonlinearities and memory effects at the same time. In this regard, proper formulation of nonlinear memory effect is at the core of DPD model design.

In existing approaches, the nonlinear memory effect is typically decomposed into different components. Some common configurations are shown in Fig. 2. The simplest decomposition is to address each memory sample separately, as in the MP model. To represent more complex nonlinear memory effect, previous studies employed a two-input nonlinear function. For example, in GMP model, each polynomial involves two adjacent samples (such as $\tilde{x}(n-1)$ and $\tilde{x}(n-2)$), while DDR model considers the instantaneous sample and the other memory sample (such as $\tilde{x}(n)$ and $\tilde{x}(n-2)$). It is worth noting that the nonlinear functions discussed here are not restricted to polynomials. Other nonlinear operators, such as LUTs [11], [30] or the combination of polynomials and LUTs [12], are also used.

Following this line of thinking, it is easy to consider using more complex nonlinear functions, such as three-input ones, if we need to model ever more complicated nonlinear memory effect, as in Fig. 3. This can be achieved by introducing more terms from the full Volterra model [31], [32] or incorporating LUTs with higher dimensions [33], [34]. Nevertheless, the size and complexity of the model can grow exponentially with the input dimension of nonlinear functions. In wideband applications, as the model involves more memory samples, such configuration will lead to prohibitive model complexity.

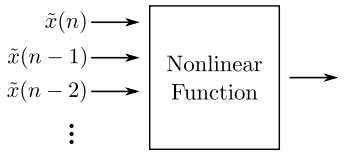

Fig. 3. Ideal memory effect representation.

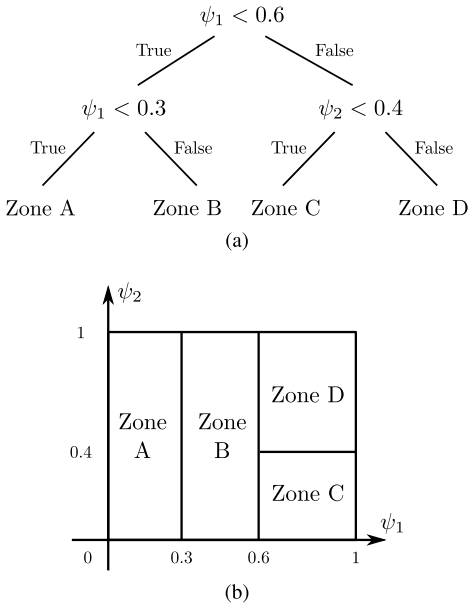

Fig. 4. Illustration of a decision tree. (a) Tree structure. (b) Zone selection.

To solve this dilemma, in this work, we attempt to move one step beyond the conventional "memory decomposition" methodology and present a new BMT approach by integrating two machine learning techniques, namely decision tree and model boosting.

III. Proposed Model Structure

Decision tree, as a classification model, can divide the input space into different zones [35]. An example of decision tree is shown in Fig. 4(a), and Fig. 4(b) shows the divided input space. The tree has two real-valued splitting features ψ_1 and ψ_2. When a new input sample $\tilde{x}(n)$ comes in, it is directed to one of the zones based on the value of splitting features $\psi_1(n)$ and $\psi_2(n)$. In this example, if $\psi_1(n) = 0.7$ and $\psi_2(n) = 0.3$, we have $\psi_1(n) \geq 0.6$ and $\psi_2(n) < 0.4$, so $\tilde{x}(n)$ will be classified to Zone C.

To use decision trees to represent PA memory effects, a simple way is to define splitting features with magnitude of the signals, i.e., $\psi_m(n) = |\tilde{x}(n-m)|$, making the model outcome dependent on both current and past memory samples.

An important feature of the decision trees is that they can perform automatic feature selection during the training process. When a decision tree is trained, it will determine the optimal splitting feature and threshold for each node in the tree. Thus, it is possible to feed a large number of memory samples to the decision tree, and we can simply let the optimization algorithm select the dominant factors and model the nonlinear relationship automatically.

However, a single decision tree will only pick a few most important features. It may not be sufficient to fully characterize the PA behavior. To cope with this deficiency, we propose to employ a model boosting technique to build multiple decision trees and encourage each tree to involve different memory samples. With a few more trees, the aggregated model is able to cover all necessary memory samples and construct the model efficiently.

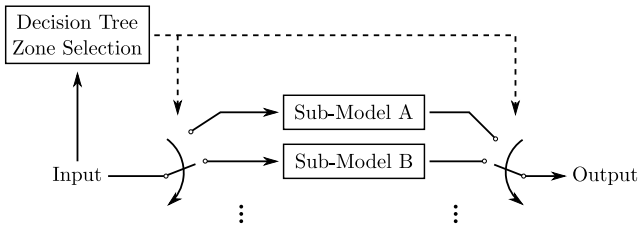

Fig. 5. Operation principle of model trees.

In the following, we will discuss the detailed construction of the proposed model by applying the abovementioned methodology.

A. Decision Tree-Based Piecewise Modeling

The decision trees need to be properly arranged in a behavioral model to unleash their full expressive power. In this work, instead of designing a model from scratch, we propose to combine decision trees with small-size pruned Volterra models. The resulting model is referred to as "model trees."

As shown in Fig. 5, a model tree defines a set of independent submodels, and each input sample selects one of the submodels for DPD processing. The process of selecting the corresponding submodel is labeled as "zone selection" and realized by using a decision tree. In other words, the decision tree acts as switches to select the best model for each data sample. For instance, in the previous example, if $\tilde{x}(n)$ belongs to Zone C, it will be processed by submodel C of the DPD. With properly designed submodels, the flexibility of the proposed method is greatly extended and it is possible to take full advantage of the prior knowledge of the PA by adopting models that are best suited for a specific PA.

Because of the piecewise structure, model trees also have low power consumption because each input data sample will only be processed by the relevant nonlinear operators, rather than the full model. Compared with using a single submodel, it greatly increases the degrees of freedom in the model and, at the same time, makes the basis functions more diverse because different submodels are naturally orthogonal to each other. Therefore, the proposed model can achieve good modeling accuracy with minimal hardware resources and power consumption.

Compared with the existing piecewise models, the proposed model tree approach has some unique and important features:

1) Supervised Threshold Optimization: As we will show in Section IV, the optimization procedures can directly optimize the splitting thresholds of all nodes in the decision tree jointly with the submodel coefficients so that the DPD can be fully optimized to achieve the best performance for the PA under test. It is very different from the conventional approaches that determine the threshold by analyzing the signal characteristics [20] or the memoryless PA behavior only [36]. Thus, unlike existing solutions, the proposed algorithm takes full

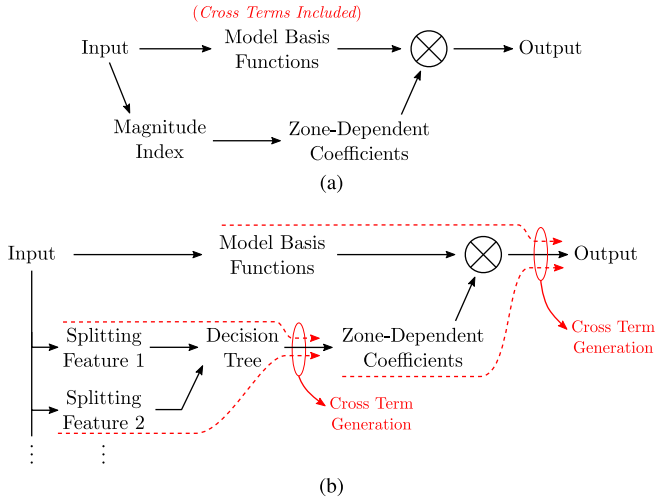

Fig. 6. Cross-term generation mechanism in (a) conventional piecewise models and (b) model trees.

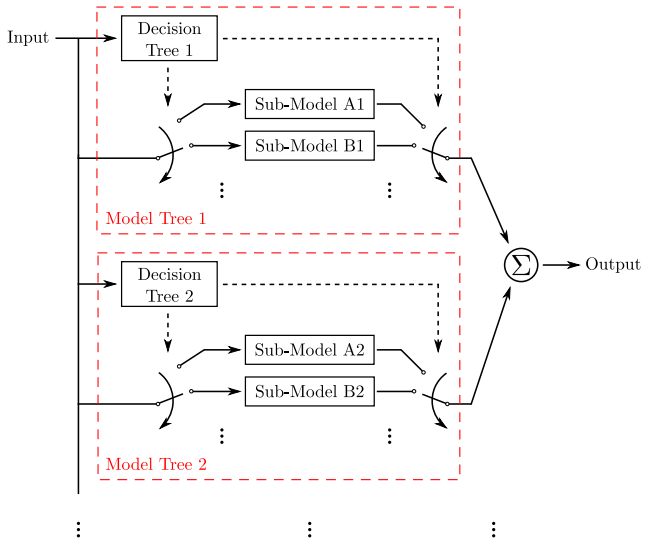

Fig. 7. Full BMT model architecture.

account of PA's memory effect and can robustly solve the threshold values within reasonable complexity budget.

2) Optimized Selection of Memory Samples: In model trees, the splitting feature of each node can be chosen freely from memory samples of the input signal. While existing methods must designate the splitting feature manually in advance, the proposed approach performs the selection automatically by the optimization algorithm, so the flexibility of the DPD model is greatly extended. Moreover, as only one feature is selected for every split, the proposed model tree can work effectively even when there are large numbers of potential features, so it can cover more memory samples than the conventional methods.

3) Modeling of Complex Memory Effect: As discussed in the literature [8], [9], cross terms, namely the interactions between current and delayed samples, play an important role in improving model accuracy and DPD performance. In conventional piecewise models that use the magnitude of the instantaneous input sample as index, the cross terms can only be embedded in the model basis functions, as shown in Fig. 6(a). In contrast, the proposed model tree method offers a flexible way of cross-term generation. As shown in Fig. 6(b), the decision tree is built with many splitting features, so different features can be naturally mixed to produce cross-term effects. Moreover, the zone-dependent coefficients are further multiplied with the model basis functions to create a more diverse mixture of nonlinear memory effects. With the supervised optimization procedures, the two cross-term generation mechanisms of model tree can be jointly optimized to generate powerful cross terms to best model the target PA behavior.

B. Model Boosting and Full Model Structure

To compensate for the distortions excited by wideband signals, the DPD model needs complex and diverse cross terms to address the memory effect. Model trees can efficiently construct such cross terms using the tree structure. However, for one model tree, only the interactions involving the selected splitting features can be modeled. Thus, a single model tree can only include limited types of cross terms.

To enhance the performance, we propose to employ multiple parallel model trees using a model boosting technique [35], [37]. The key idea of model boosting is to aggregate many simple basic models to improve modeling accuracy. In this way, the modeling residue from one model can be compensated by the remaining models.

By applying the boosting technique to model trees, we obtain the BMT model. In the BMT model, we construct multiple model trees and let each model tree have distinct tree structure and submodel coefficients. As each model tree has a unique tree structure, we can integrate more different cross terms into the model. It is worth mentioning that using multiple trees has a very different effect than using more zones in a single tree because every new tree involves different memory samples and builds new types of cross terms, while increasing the number of zones alone is only similar to increasing the nonlinear order.

The architecture of the full BMT model is shown in Fig. 7. The input signals are fed to different decision trees and the tree structures make individual decisions on zone selection for each model tree. The input data are then processed by the corresponding submodels and each model tree generates one output signal which is later summed up to produce the final output of the model. While the full model has many blocks, it can be implemented efficiently in hardware, as will be detailed in Section III-C.

C. Implementation of BMT

In this work, as a proof of concept, we use the magnitude of current and past input samples, $|\tilde{x}(n-m)|$, as splitting features and employ pruned Volterra models as submodels. As the calculation of $|\tilde{x}(n-m)|$ or $|\tilde{x}(n-m)|^2$ is included in most pruned Volterra models, there is virtually no additional complexity to generate the splitting features. Moreover, the splitting features have a much lower accuracy requirement than other parts of the DPD model because we only need to know which zone the sample lies in and do not need to know the exact value. Thus, depending on the application, this feature may be exploited to optimize the hardware utilization.

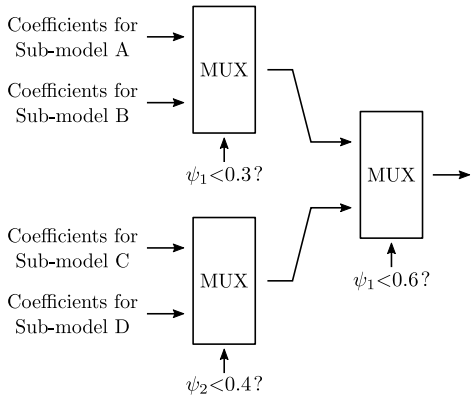

Fig. 8. Example hardware implementation of decision tree structures.

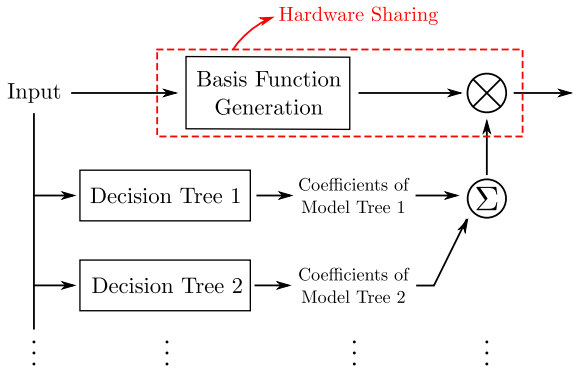

Fig. 9. BMT implementation with hardware sharing.

The core feature of the BMT model, i.e., the tree structure, can be implemented with very low hardware complexity because the design of this part involves comparators and multiplexers only. An example is shown in Fig. 8, which implements the tree structure shown in Fig. 4(a).

The implementation of the full BMT model can be further simplified because different arithmetic operations can be shared in hardware implementation if the same model expression is employed in all submodels. The first shared component is the basis function generation block since the same basis functions can be fed to all parts of the model. In addition, in each model tree, instead of switching the submodels, we can just switch the coefficients, as shown in Fig. 8. Another hardware simplification is the sharing of coefficient multipliers between different model trees. Specifically, we can add the coefficients from different model trees together before multiplying them with the model basis functions, so the complex multiplications between model coefficients and basis functions can be shared.

As shown in Fig. 9, due to the proposed hardware sharing techniques, the additional complexity for adding a new tree is reduced to adders (for summing up coefficients) and multiplexers (for tree structures) only, and no complex multiplier is needed. Note that one complex multiplier can consume over 1000 LUTs on FPGA [38], whereas adders and multiplexers only need fewer than 100 LUTs [39], [40]. Thus, by utilizing low-complexity components, the final model to be implemented can have very low hardware utilization.

Therefore, the BMT model derived in this section shows great potential to achieve both powerful linearization performance and low hardware complexity at the same time.

IV. Model Training and Coefficient Extraction

After deriving the model structure, we now present how to extract the model parameters. The main framework of the model training is shown in Fig. 10. The training follows a stepwise strategy, and the trees are trained one by one. As shown in the left of Fig. 10, the detailed training process can be described as follows.

1) Add a new empty tree to the model.
2) Optimize every node of the new tree using a top-down optimization strategy and a binary-split alternate minimization (BAM) algorithm.
3) Fine-tune the coefficients of all existing submodels.
4) Update the target signal for the next model tree.
5) If more trees are needed, go to step 1 and continue the optimization.

We will discuss each step in detail as follows.

To facilitate the discussion, we make the following definitions. All input and output samples are collected in vector $\mathbf{x}$ and $\mathbf{y}$, respectively. Ψ is a matrix where each column is a splitting feature, and Φ is a matrix where each column is a basis function of submodel.

A. Construction of Tree Structures

To initialize the training algorithm, the modeling target for building the first tree is the desired model output signal $\mathbf{y}$. Starting from the second tree, the target signal is the modeling residue from the previous tree. With the proposed training procedure, each model tree will be fit to the target signal.

To properly construct the tree, we employ a layer-by-layer procedure, which starts from the top node and gradually moves to nodes in lower layers. As shown in the middle of Fig. 10, to optimize the ith node, we first treat the node as an end node and assume that it has no child nodes. The splitting feature and threshold of this node (v_i and τ_i) is then optimized by the BAM algorithm presented in the right of Fig. 10. The BAM algorithm will be introduced in Section IV-B. Once the optimal splitting feature and threshold are found, the dataset is divided into two parts based on the optimal split and fed to its two child nodes in the layer below. The optimization of the child nodes can thus start by using a subset of the data. This process is iterated until all nodes of the tree are built.

It is worth mentioning that once optimized, the splitting feature and threshold of a given node will be fixed and are not affected by the training of nodes in lower layers. In addition, within one layer, the optimization of different nodes can be parallelized for faster execution speed on certain hardware.

B. Binary-Split Alternate Minimization Algorithm

Different methods to determine threshold values have been presented in the prior art, including uniform spacing [12], k-means clustering [20], and memoryless nonlinearity analysis [13], [36]. Nevertheless, since they do not consider the memory effects of the PAs and require manual selection

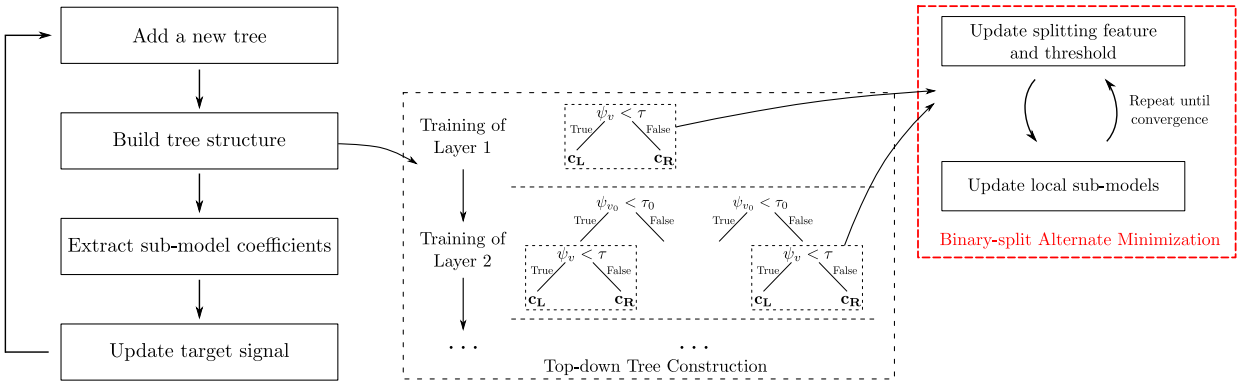

Fig. 10. Proposed training framework for BMT model.

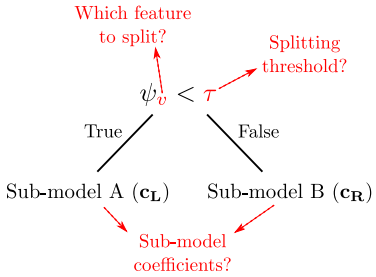

Fig. 11. Optimizable parameters in a single model tree with only one node.

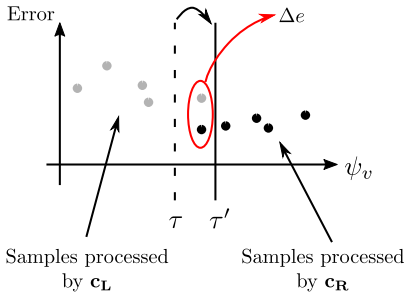

Fig. 12. Illustration of splitting threshold optimization.

of splitting features, their performance may be limited in wideband DPD scenarios.

To find the optimum splitting features and thresholds, the training process must jointly optimize both the splitting criterion and submodel coefficients. An intuitive model training method is to do an exhaustive search of all possible splitting features and splitting thresholds [41], [42]. For each candidate split, we extract the submodel coefficients with least-squares (LS) and calculate the total squared error for both sides. The best split will be found after sweeping over all features and thresholds. While this method is widely adopted [43], [44], it has prohibitively high computational complexity because we need to solve a large number of LS problems.

To reduce computational complexity, a novel BAM algorithm is proposed. As shown in Fig. 11, the splitting criterion to be optimized is $\psi_v < \tau$, where v is the index of the feature to be split and τ is the splitting threshold, so both v and τ are optimizable. The submodel coefficients for the two sides are $\mathbf{c_L}$ and $\mathbf{c_R}$. As shown in the right of Fig. 10, the BAM algorithm alternately optimizes the splitting criterion and submodel coefficients.

In the first step, the submodel coefficients are fixed, and we optimize the splitting criterion. In this case, wherever the splitting threshold is, each data sample must belong to one of the two submodels, that is to say, an arbitrary input sample $\tilde{x}(n)$ must be processed by either $\mathbf{c_L}$ or $\mathbf{c_R}$. Suppose that the modeling error with $\mathbf{c_L}$ and $\mathbf{c_R}$ is $e_L(n)$ and $e_R(n)$, respectively. As shown in Fig. 12, if the splitting threshold is moved by just a little such that $\tilde{x}(n)$ is processed by $\mathbf{c_R}$ before moving and by $\mathbf{c_L}$ after moving, the variation of total error $\Delta e(n)$ will be $e_L(n) - e_R(n)$. The modeling error with a given splitting criterion $\psi_v < \tau$ can thus be obtained by accumulating Δe of all samples below the threshold. Therefore, we can evaluate all possible splitting thresholds simply by moving the threshold value across the training data. The split with the lowest error will be chosen for the current node, after sweeping over all potential splitting features.

In the second step, the splitting threshold is fixed, so the dataset can be divided into two parts. We can then update the coefficients of the two submodels using LS.

By iterating between the two steps, the splitting criterion and submodel coefficients can be jointly optimized. The complete description of the BAM algorithm, as well as the implementation details, is presented in the Appendix.

C. Submodel Coefficient Extraction

Once the tree structures are determined by the top-down tree construction strategy and BAM algorithm, we can extract the submodel coefficients of all model trees.

With tree structure fixed, the model becomes linear in the submodel coefficients and can be expressed in the matrix format. For example, suppose that in the tth model tree, we have $\tilde{x}(1)$, $\tilde{x}(3)$, and $\tilde{x}(4)$ in the first zone and $\tilde{x}(2)$ in the second zone. Then, the model matrix can be represented as

$$\mathbf{X_t} = \begin{bmatrix} \Phi(1) & \mathbf{0} \\ \mathbf{0} & \Phi(2) \\ \Phi(3) & \mathbf{0} \\ \Phi(4) & \mathbf{0} \\ \vdots & \vdots \end{bmatrix} \tag{1}$$

where $\Phi(n)$ is nth row of Φ and $\mathbf{0}$ represents an all-zero row vector with the same size as $\Phi(n)$. Subsequently, we concatenate the model matrices of all existing model

trees, including both current and previous ones

$$\mathbf{X} = [\mathbf{X_1}, \mathbf{X_2}, \ldots]. \tag{2}$$

Coefficients of all model trees can then be extracted by LS

$$\hat{\mathbf{c}} = \left(\mathbf{X}^H\mathbf{X}\right)^{-1}\mathbf{X}^H\mathbf{y} \tag{3}$$

where $\hat{\mathbf{c}}$ is a vector including all submodel coefficients.

Afterward, the target signal for tree construction is updated to be the current modeling residue

$$\mathbf{r} = \mathbf{y} - \mathbf{X}\hat{\mathbf{c}}. \tag{4}$$

That is, the residue of the current BMT model is used as the target for building the next tree.

This process is iterated until all trees are built, leading to an optimized BMT model.

D. Dynamic Adaptation in Real Time Operation

After the DPD system is deployed for real-world operation, the DPD model needs to be updated occasionally to track the variation of PA characteristics. If the variation of PA behavior is not significant, the model tree structure can be fixed and only the submodel coefficients need to be updated. In this case, the estimation can be done with (3), which is the same as conventional LS method, leading to reduced computational complexity.

V. Results

A. Experimental Setup and Evaluation Metrics

To validate the model performance, a test platform was set up, as shown in Fig. 13, which includes PC, signal generator, driver amplifier, PA, attenuator, and spectrum analyzer. The PA under test was an in-house designed broadband gallium nitride (GaN) load-modulated balanced PA operating at 2.1 GHz with 37-dBm output power and 42% drain efficiency [28]. The excitation input signals were five-carrier 100-MHz and ten-carrier 200-MHz orthogonal frequency-division multiplexing (OFDM) signals, both with 8-dB peak-to-average power ratio (PAPR). The sampling rates were set to four times the signal bandwidths. Recorded I/Q input and output samples were time-aligned and normalized before training the model. The model extraction and predistorted signal generation were both performed in MATLAB.

In the forward modeling case, the normalized mean square error (NMSE) is used as the indicator for modeling accuracy. In the DPD test, we employ both NMSE and adjacent channel power ratio (ACPR) as performance metrics. For both types of evaluation, 80 000 samples were used for model extraction and another set of 80 000 data points was used for performance evaluation.

In the complexity analysis, we follow the same methodology as in [45]. The main metric is the number of floating-point operations per sample (FLOPs). Also, the complexity of submodels and zone selection is both included.

To evaluate the proposed BMT method, we compared the proposed BMT technique with both conventional Volterra models and other piecewise modeling methods, including

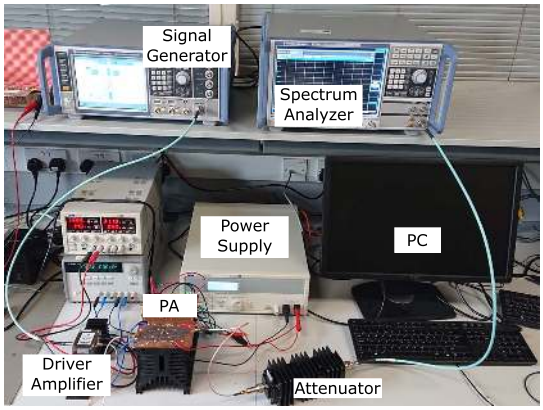

Fig. 13. Photograph of the DPD test bench.

VS [20], MLUT [34], and DLUTI [12] methods. In all piecewise models, the submodels had the same model structure as the pruned Volterra model but with different parameter settings. The model extraction methods for these methods followed the same procedures as the original papers.

In the evaluation, we used different pruned Volterra models to best suit the specific cases. For the 100-MHz test, we used GMP model without leading cross terms as submodels. Both even- and odd-order terms were used. When increasing the signal bandwidth to 200 MHz, the memory effect becomes stronger, so an additional type of DDR term, $|\tilde{x}(n)|^{p-3}\tilde{x}^2(n)\tilde{x}^*(n-m)$, was added to the model.

B. Forward Modeling Performance

The modeling accuracy of the proposed BMT model is first evaluated in a forward modeling setup.

1) Performance Comparison With Parametric Sweeps: To systematically draw a comparison between the proposed method and the conventional models, we performed a parametric sweep over the model parameter settings and evaluated the best performance/complexity tradeoff for each model.

In the sweep using 100-MHz signals, we compared our BMT model with GMP, VS-GMP, DLUTI-GMP, and MLUT-GMP models. The swept parameters of GMP model included memory depth M, polynomial order P, and cross-term length L. Specifically, we swept P from 2 to 8, changed M from 3 to 6, and varied L from 0 to 2. In all piecewise models, we used the same sweeping range for M and L. Due to the piecewise operation, using high nonlinear order is usually unnecessary, so we only considered $P = 1$ or 2. In VS and DLUTI models, the number of zones K had the value of 2, 4, 8, or 16. For the MLUT model, the same settings were used except for the number of zones. We considered six cases, including three 1-D cases ($K_0 = 4$, 8, or 16), two 2-D cases ($K_0 = K_1 = 4$ and $K_0 = 8$ and $K_1 = 2$), and a 3-D case ($K_0 = 4$, $K_1 = 2$, and $K_2 = 2$), where K_i refers to the number of zones in dividing $|\tilde{x}(n-i)|$. The sweep of the BMT model had the same sweeping range for P, M, and L as other piecewise models. N_s could be 2 or 3 and N_t was swept from 1 to 3.

For 200-MHz signals, most settings were kept the same, except that DDR terms were added and M was swept

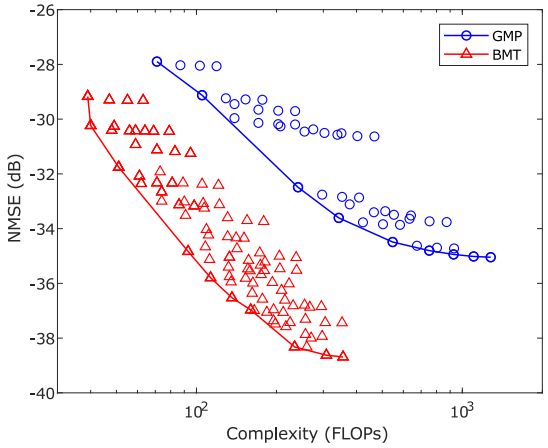

Fig. 14. NMSE performance of GMP and BMT models with sweeping parameters.

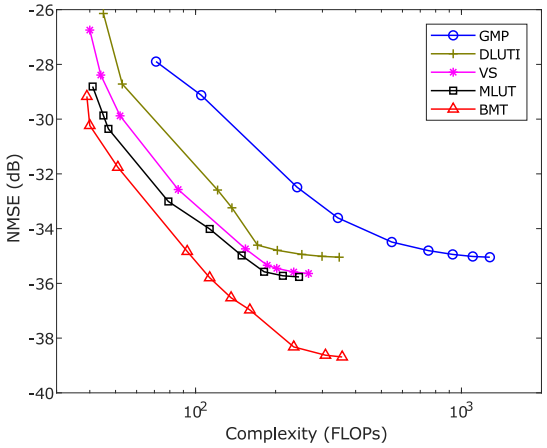

Fig. 15. Performance comparison between different models using 100-MHz signals.

from 4 to 8 instead. The two modifications were made to address the wider bandwidth.

To give a quick impression of the results of the parametric sweep, we plot the performance of all model settings for GMP and BMT models under 100-MHz excitation in Fig. 14, where each marker represents the performance and complexity for one model configuration. The solid lines in the plot represent the best performance/complexity tradeoff for the models. The comparison shows that the proposed BMT model significantly outperforms the conventional GMP model in terms of both modeling accuracy and computational complexity.

The full sweeping results for the 100- and 200-MHz cases are shown in Figs. 15 and 16, respectively. For clarity, only the best results for each model are shown.

It can be seen that, in both cases, the proposed BMT model can always achieve better performance than all other models with similar complexity. Moreover, it also increases the maximum achievable accuracy by up to 4 dB, compared with the conventional Volterra models. In contrast, the achievable performance of all other piecewise models is only slightly improved by around 1 dB. Thus, it clearly shows that the proposed BMT method not only uses the hardware resources more efficiently but also successfully builds the desired cross terms that are missing in existing methods.

2) Performance Comparison With Increasing Number of Zones: To more clearly demonstrate the advantage of the BMT technique over the existing methods, we used the same submodel for all piecewise models and see how they perform when we gradually increase the number of zones.

The parameters of the submodels were set to $P = 2$, $L = 2$, and $M = 6$ (for 100-MHz signals) or 8 (for 200-MHz signals). For MLUT and BMT models where different configurations may have the same number of zones, the best NMSE result is displayed.

The full results are shown in Figs. 17 and 18 for 100- and 200-MHz test signals, respectively. It shows that the BMT technique has superior performance than all other methods. In particular, when using over ten zones, the BMT model can outperform the best competitor by at least 1 dB in both

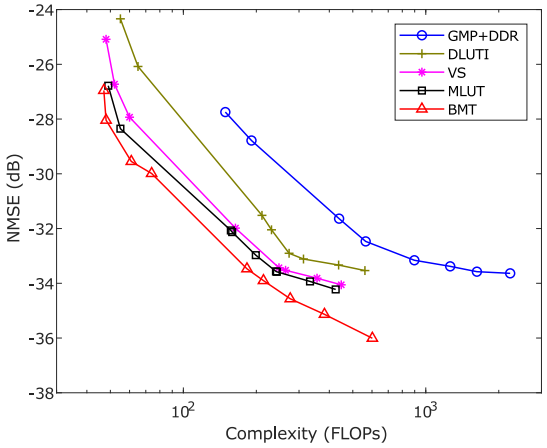

Fig. 16. Performance comparison between different models using 200-MHz signals.

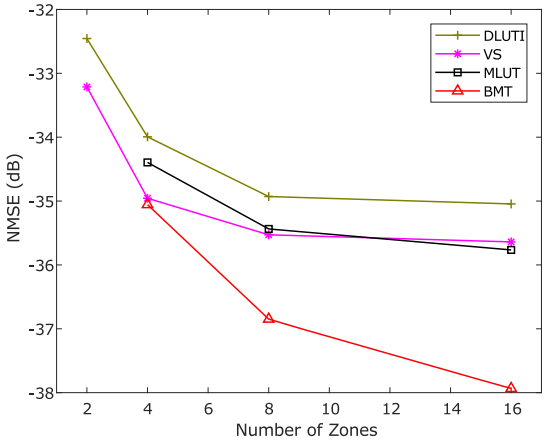

Fig. 17. Performance for different piecewise techniques with the same submodel using 100-MHz signals.

cases, and its lead is still enlarging with increasing number of zones. Actually, the performance of other methods has almost saturated in this case. It clearly shows that the proposed BMT method can construct more effective cross terms and achieve better performance using the same submodel.

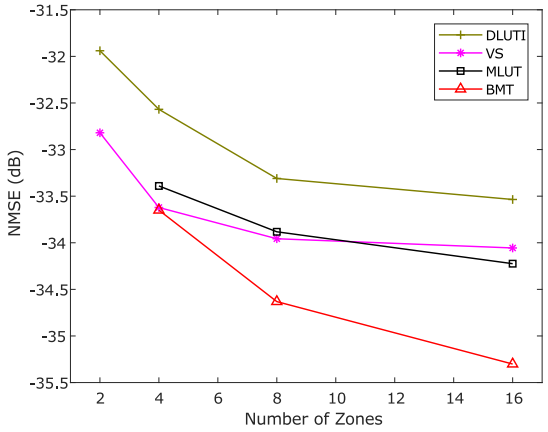

Fig. 18. Performance for different piecewise techniques with the same submodel using 200-MHz signals.

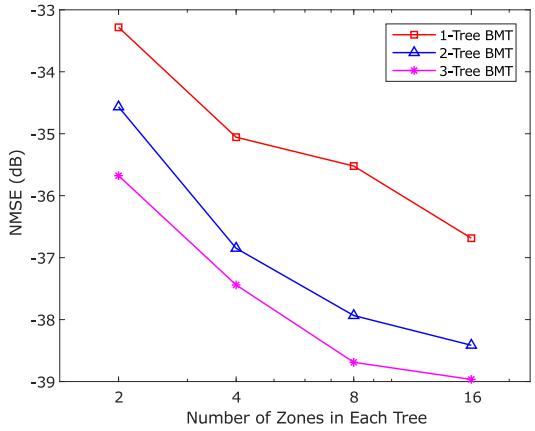

Fig. 19. Performance of BMT models with different configurations using 100-MHz signals.

3) Influence of BMT Model Configurations: Afterward, we further investigate the influence of BMT model configurations. Using the same submodel as the previous comparison, we swept over the number of trees and the number of tree layers of the model. The results for 100- and 200-MHz signals are shown in Figs. 19 and 20, respectively. The general trend of the two cases is very similar.

According to the results, the performance of BMT model is boosted significantly by employing multiple trees. A key observation is that the combination of piecewise and parallel structures can provide decent performance improvement, which is difficult to achieve with the piecewise modeling technique alone. For example, in Fig. 19, when we use four zones in each tree, the NMSE is improved from -35 to -37 dB by adding the second tree. This performance is even better than what we can obtain with 16 zones if we keep using only one tree. Moreover, the performance improvement slows down when the number of zones in a tree increases to a large number. Thus, to further improve the accuracy, employing additional trees provides a new dimension to explore. Therefore, to achieve good linearization performance, it is important to take advantage of both piecewise and parallel structures of the model.

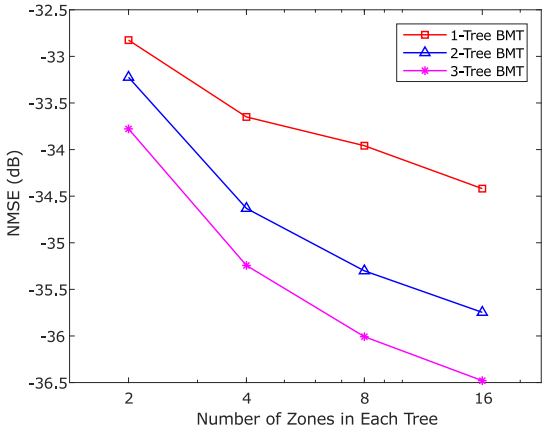

Fig. 20. Performance of BMT models with different configurations using 200-MHz signals.

TABLE I
COMPARISON OF DIFFERENT MODELS USING 100-MHZ SIGNALS

Model	FLOPs	NMSE (dB)	ACPR (dBc)
w/o DPD	N/A	-22.5	-30.3/-30.5
GMP	803	-34.9	-41.3/-43.2
DLUTI	251	-34.9	-41.2/-43.4
VS	218	-35.5	-42.0/-43.6
MLUT	211	-36.0	-41.9/-43.4
BMT	234	-38.4	-45.5/-46.2

C. DPD Performance Comparison

To ultimately verify the performance of the proposed method, we evaluated the linearization performance of the models using DPD tests. During all tests, closed-loop estimation was used to extract the model coefficients.

In 100-MHz tests, we set the GMP parameters to $P = 6$, $M = 5$, and $L = 2$. For piecewise models, the polynomial order of submodels was changed to $P = 2$. This configuration has shown to achieve good performance for all methods in forward modeling. Both DLUTI and VS used eight zones because using more zones had little improvement on accuracy. MLUT method used 2-D LUTs with $K_0 = K_1 = 4$, which is the best setting in forward modeling. In the BMT model, we set $N_s = 3$ and $N_t = 3$. To better balance the complexity, the submodel used in the BMT method had $L = 1$. Thus, all piecewise models had around 200 FLOPs.

It can be shown that the BMT achieves leading linearization performance. Spectral results using 100-MHz signals are shown in Fig. 21. The AM-AM and AM-PM curves for the BMT model in this case are shown in Fig. 22. The spectral and time-domain AM-AM results show that the PA exhibits a significant memory effect, but the proposed method can still offer good linearization performance. The detailed performance and complexity results are presented in Table I. It shows the BMT method clearly outperforms all existing methods by a large margin with similar level of complexity, which agrees with the results of forward modeling.

To compare the performance of the models with wider bandwidth, we further used a 200-MHz setup. In this

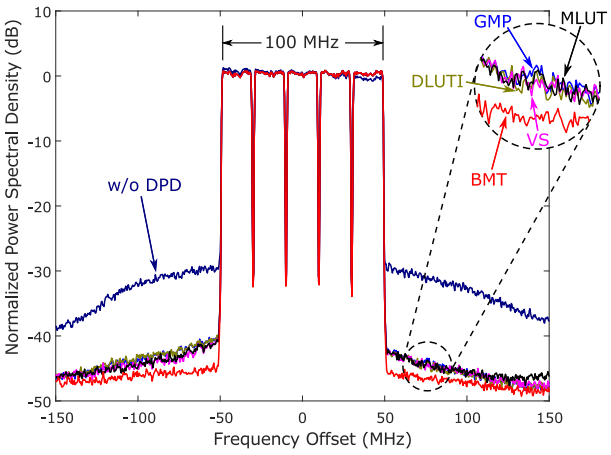

Fig. 21. Spectral results for different models using 100-MHz signals.

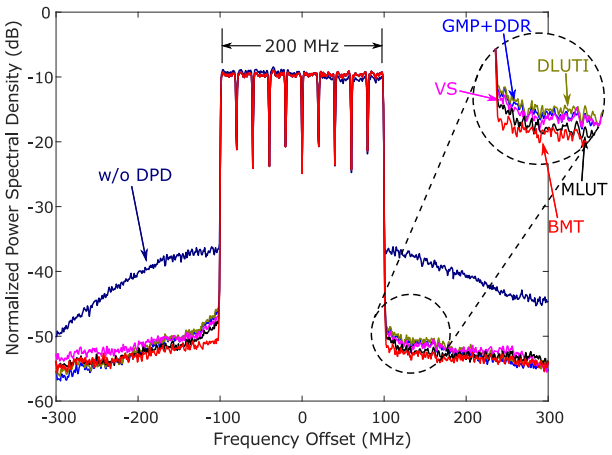

Fig. 23. Spectral results for different models using 200-MHz signals.

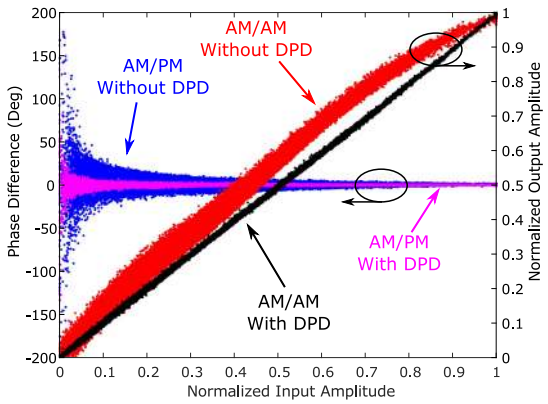

Fig. 22. AM-AM and AM-PM characteristics with and without BMT-DPD from the 100-MHz test.

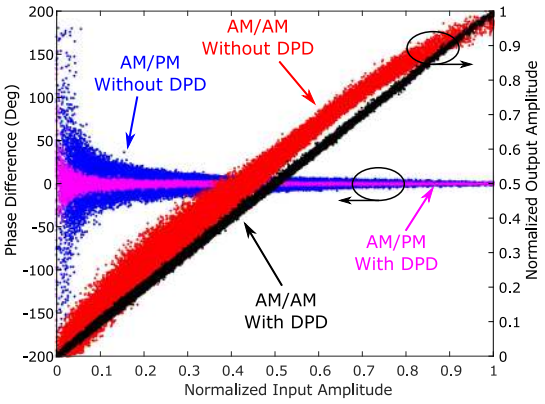

Fig. 24. AM-AM and AM-PM characteristics with and without BMT-DPD from the 200-MHz test.

TABLE II
COMPARISON OF DIFFERENT MODELS USING 200-MHZ SIGNALS

Model	FLOPs	NMSE (dB)	ACPR (dBc)
w/o DPD	N/A	-20.1	-27.1/-27.0
GMP+DDR	1255	-32.7	-37.4/-39.5
DLUTI	415	-32.5	-37.0/-39.2
VS	358	-32.8	-38.1/-40.0
MLUT	351	-33.9	-38.9/-40.8
BMT	382	-35.6	-41.0/-41.8

case, compared with 100-MHz tests, memory depth M was increased to 8 and additional DDR terms were used. The rest settings were the same as the 100-MHz case. The selected model configurations all perform well in forward modeling and the complexities are around 400 FLOPs.

The detailed spectral performance for selected model settings in the 200-MHz test is shown in Fig. 23, and we also depict the AM-AM and AM-PM characteristics of BMT model in Fig. 24. We draw a detailed comparison of the different models for the 200-MHz test in Table II. The results of this test show a similar trend as the previous 100-MHz test and further confirm the performance of the proposed BMT approach for wideband behavioral modeling and linearization.

It is worth mentioning that, while the proposed model is mainly developed for the wideband frequency-division duplexing (FDD) systems, it can also be applied to other cases, such as time-division duplexing (TDD) systems, but some special treatments may be required. For instance, transient memory effects in TDD must be considered. Nevertheless, the combination of our work and other existing research on TDD linearization techniques [46], [47] can also potentially lead to good linearization performance.

D. Model Extraction Complexity

In this section, we evaluated different model training strategies for BMT. Specifically, we compared the proposed BAM algorithm with the conventional exhaustive search method [42] mentioned in Section IV-B. To complete the search within a reasonable time, we tested eight thresholds for each splitting feature.

The training time and accuracy are reported in Table III, and the training was performed on a PC. It shows that the proposed BAM method can extract the full model with similar accuracy and significantly shorter time than the exhaustive search strategy. The results show that the computational complexity of full model update is still manageable in the dynamic environment.

It is also worth noting that, as discussed in Section IV-D, it is possible to update only the submodel coefficients of the

TABLE III
COMPARISON OF DIFFERENT MODEL TRAINING METHODS

Method	Execution Time (s)	NMSE (dB)
Exhaustive Search	225.2	-38.0
BAM	20.3	-37.8

BMT model during the dynamic adaptation. In our test, when PA behavior does not vary much, such a strategy can further reduce the model extraction time to 4.5 s while maintaining the same level of linearization performance.

Furthermore, as we target at wideband DPD solutions, the bandwidth limit for data converters is also a practical issue in real-world applications [4]. Some recent works have focused on this problem and proposed some promising solutions [48], [49]. We believe that the combination of this work with these techniques can be a good topic for future research.

VI. CONCLUSION

In this article, we present a novel BMT model to achieve improved linearization performance under wideband modeling scenarios with low hardware complexity. The proposed approach extends the existing DPD model architectures by introducing a combination of piecewise and parallel modeling techniques. With the help of both decision tree-based piecewise structure and boosting-based parallel architecture, the proposed BMT model is shown to enhance the modeling capability of conventional pruned Volterra models with low complexity. We also develop dedicated model training algorithms to efficiently identify the BMT model structure and coefficients. Therefore, we believe that the BMT modeling approach offers a new and promising solution to address the performance and complexity challenges in the design of future DPD models.

APPENDIX
IMPLEMENTATION OF BINARY-SPLIT ALTERNATE MINIMIZATION ALGORITHM

In this appendix, we present the detailed implementation of the proposed BAM algorithm. The splitting criterion to be optimized is $\psi_\upsilon < \tau$, where υ is the index of the feature to be split and τ is the splitting threshold. Input samples below the threshold are collected in $\mathbf{x_L}$, and the corresponding output samples, entries in regression matrix and submodel coefficients, are $\mathbf{y_L}$, $\Phi_\mathbf{L}$ and $\mathbf{c_L}$, respectively. Similarly, all entities for representing samples above the threshold will have the subscript R. Note that, starting from the second tree, the target signal becomes the residue of the previous tree, so $\mathbf{r}$ is used as target signal instead of $\mathbf{y}$ in this case.

The BAM method alternately optimizes the tree structure and submodel coefficients with two iterative steps.

1) Step I: In step 1, the submodel coefficients are fixed, and we aim to find an optimum splitting threshold. As discussed in Section IV-B, after fixing submodel coefficients, an arbitrary input sample $\tilde{x}(n)$ must be processed by either $\mathbf{c_L}$ or $\mathbf{c_R}$. Thus, the modeling error for this sample can only have two possible values, $|\tilde{y}(n) - \Phi(n)\mathbf{c_L}|^2$ or $|\tilde{y}(n) - \Phi(n)\mathbf{c_R}|^2$, where $\tilde{y}(n)$ is the target and $\Phi(n)$ is nth row of Φ. Therefore, if the splitting threshold is moved by just a little such that $\tilde{x}(n)$ is processed by $\mathbf{c_R}$ before moving and by $\mathbf{c_L}$ after moving, the variation of total error can be expressed by

$$\Delta e(n) = |\tilde{y}(n) - \Phi(n)\mathbf{c_L}|^2 - |\tilde{y}(n) - \Phi(n)\mathbf{c_R}|^2. \quad \text{(A.1)}$$

Based on this observation, we can evaluate all possible splitting thresholds efficiently by sweeping over all training data using (A.1). To be specific, when sweeping over the pth splitting feature, we first sort the data samples based on the value of ψ_p and then gradually move the splitting threshold from min ψ_p to max ψ_p. Every single move will only change the modeling error of one training sample. We define $e(p, n)$ as the total modeling error when the splitting threshold is $\Psi(n, p)$. If we set the initial error to zero, i.e., $e(p, 0) = 0$, the relative modeling error of all possible splitting thresholds can be obtained by

$$e(p, n) = e(p, n-1) + \Delta e(n). \quad \text{(A.2)}$$

Finally, the split with the lowest error will be chosen for the current node.

In practice, it is also important to avoid too imbalanced split, i.e., the number of samples for one branch is below a predetermined threshold N_{th}. When the number of samples is too small, the estimation of submodel coefficients may not be accurate, resulting in overfitting. Besides, it is also not efficient to assign a submodel that works only for very few samples. Thus, too imbalanced split should be avoided and excluded before searching for the best split. The final results are generally not sensitive to the value of N_{th}. In this work, we set $N_{\text{th}} = 5Q$, where Q is the number of coefficients in a submodel.

2) Step II: In step 2, the splitting threshold is fixed. Thus, the dataset can be divided into two parts based on the splitting threshold obtained from the previous step. The model becomes linear in parameters, and the submodel coefficients can be easily extracted using LS

$$\hat{\mathbf{c}}_\mathbf{L} = \left(\Phi_\mathbf{L}{}^H \Phi_\mathbf{L}\right)^{-1} \Phi_\mathbf{L}{}^H \mathbf{y_L} \quad \text{(A.3)}$$

and

$$\hat{\mathbf{c}}_\mathbf{R} = \left(\Phi_\mathbf{R}{}^H \Phi_\mathbf{R}\right)^{-1} \Phi_\mathbf{R}{}^H \mathbf{y_R}. \quad \text{(A.4)}$$

3) Complete BAM Algorithm: The two steps will be iterated until convergence. Based on our experience, typically, around five iterations are sufficient to achieve satisfactory results. As the main complexity, LS extraction, is performed only once in every iteration, the computational complexity and memory requirement can be kept low.

To initialize the algorithm, we need to have initial coefficients for the two submodels. We achieve this by dividing the data into two clusters and extracting one submodel in each cluster. In our implementation, the clustering is done by performing the k-means algorithm on the target signal $\mathbf{y}$ (or $\mathbf{r}$) with $k = 2$. In this way, we obtain two submodels that are different from each other but can both reflect the PA characteristics. One thing to note is that the proposed

Algorithm 1 BAM Algorithm

Input: Φ, Ψ, $\mathbf{y}$

Output: υ, τ

Initialization with k-means algorithm and sub-model LS extraction
repeat
  Calculate Δe using (A.1)
  for $p = 1$ to P **do**
    Initialization: $e(p, 0) = 0$
    Sort Δe based on the pth column of Ψ
    for $n = 1$ to N **do**
      Apply (A.2) to obtain $e(p, n)$
    end for
  end for
  $(\upsilon, t) = \arg\min_{p,n} e(p, n)$ s.t. $N_{th} < n < N - N_{th}$
  $\tau = \Psi(t, p)$
  Divide samples into two parts based on the criterion $\psi_\upsilon < \tau$
  Extract sub-models using (A.3) and (A.4)
until the error converges or maximum number of iterations is reached

initialization algorithm generates two sets of coefficients, but it is unclear which one should be assigned to $\mathbf{c_L}$ (or $\mathbf{c_R}$). Thus, for better robustness, it is desirable to compare both assignments and keep the one resulting in better performance.

To implement the BMT model in practical applications, the splitting threshold τ may need to be quantized to save hardware resources. Thus, at the end of optimization algorithm, we can optionally quantize τ to a relatively low precision. Based on our experience, there is only negligible accuracy loss with up to 5-bit quantization.

A complete description of the BAM algorithm is presented in Algorithm 1.

REFERENCES

[1] 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.

[2] J. Wood, *Behavioral Modeling and Linearization of RF Power Amplifiers*. Norwood, MA, USA: Artech House, 2014.

[3] 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.

[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] T. Qi and S. He, "Power up potential power amplifier technologies for 5G applications," *IEEE Microw. Mag.*, vol. 20, no. 6, pp. 89–101, Jun. 2019.

[6] W. Cao, Y. Li, and A. Zhu, "Magnitude-selective affine function based digital predistorter for RF power amplifiers in 5G small-cell transmitters," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Jun. 2017, pp. 1539–1541.

[7] 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.

[8] 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.

[9] 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.

[10] K. J. Muhonen, M. Kavehrad, and R. Krishnamoorthy, "Look-up table techniques for adaptive digital predistortion: A development and comparison," *IEEE Trans. Veh. Technol.*, vol. 49, no. 5, pp. 1995–2002, May 2000.

[11] P. L. Gilabert, A. Cesari, G. Montoro, E. Bertran, and J.-M. Dilhac, "Multi-lookup table FPGA implementation of an adaptive digital predistorter for linearizing RF power amplifiers with memory effects," *IEEE Trans. Microw. Theory Techn.*, vol. 56, no. 2, pp. 372–384, Oct. 2008.

[12] A. Molina, K. Rajamani, and K. Azadet, "Digital predistortion using lookup tables with linear interpolation and extrapolation: Direct least squares coefficient adaptation," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 3, pp. 980–987, Mar. 2017.

[13] S. N. Ba, K. Waheed, and G. T. Zhou, "Optimal spacing of a linearly interpolated complex-gain LUT predistorter," *IEEE Trans. Veh. Technol.*, vol. 59, no. 2, pp. 673–681, Feb. 2010.

[14] T. Magesacher, P. Singerl, and M. Mataln, "Optimal segmentation for piecewise RF power amplifier models," *IEEE Microw. Wireless Compon. Lett.*, vol. 26, no. 11, pp. 909–911, Nov. 2016.

[15] 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.

[16] F. M. Barradas, T. R. Cunha, P. M. Lavrador, and J. C. Pedro, "Polynomials and LUTs in PA behavioral modeling: A fair theoretical comparison," *IEEE Trans. Microw. Theory Techn.*, vol. 62, no. 12, pp. 3274–3285, Dec. 2014.

[17] M. Younes, O. Hammi, A. Kwan, and F. M. Ghannouchi, "An accurate complexity-reduced 'PLUME' model for behavioral modeling and digital predistortion of RF power amplifiers," *IEEE Trans. Ind. Electron.*, vol. 58, no. 4, pp. 1397–1405, May 2010.

[18] 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.

[19] F. M. Barradas, L. C. Nunes, J. C. Pedro, T. R. Cunha, P. M. Lavrador, and P. M. Cabral, "Accurate linearization with low-complexity models using cascaded digital predistortion systems," *IEEE Microw. Mag.*, vol. 16, no. 1, pp. 94–103, Feb. 2015.

[20] S. Afsardoost, T. Eriksson, and C. Fager, "Digital predistortion using a vector-switched model," *IEEE Trans. Microw. Theory Techn.*, vol. 60, no. 4, pp. 1166–1174, Apr. 2012.

[21] M. Younes, A. Kwan, M. Akbarpour, M. Helaoui, and F. M. Ghannouchi, "Two-dimensional piecewise behavioral model for highly nonlinear dual-band transmitters," *IEEE Trans. Ind. Electron.*, vol. 64, no. 11, pp. 8666–8675, Nov. 2017.

[22] 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.

[23] J. Pang, Z. Dai, Y. Li, M. Li, and A. Zhu, "Multiband dual-mode Doherty power amplifier employing phase periodic matching network and reciprocal gate bias for 5G applications," *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 6, pp. 2382–2397, Jun. 2020.

[24] H. Lyu and K. Chen, "Balanced-to-Doherty mode-reconfigurable power amplifier with high efficiency and linearity against load mismatch," *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 5, pp. 1717–1728, May 2020.

[25] J. Xia, W. Chen, F. Meng, C. Yu, and X. Zhu, "Improved three-stage Doherty amplifier design with impedance compensation in load combiner for broadband applications," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 2, pp. 778–786, Feb. 2019.

[26] D. J. Shepphard, J. Powell, and S. C. Cripps, "An efficient broadband reconfigurable power amplifier using active load modulation," *IEEE Microw. Wireless Compon. Lett.*, vol. 26, no. 6, pp. 443–445, Jun. 2016.

[27] P. H. Pednekar, E. Berry, and T. W. Barton, "RF-input load modulated balanced amplifier with octave bandwidth," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 12, pp. 5181–5191, Dec. 2017.

[28] J. Pang, C. Chu, Y. Li, and A. Zhu, "Broadband RF-input continuous-mode load-modulated balanced power amplifier with input phase adjustment," *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 10, pp. 4466–4478, Oct. 2020.

[29] X. Wei, W. Chen, X. Liu, L. Chen, and F. M. Ghannouchi, "A methodology and a metric for the assessment of the linearizability of broadband nonlinear Doherty power amplifiers," *IEEE Microw. Wireless Compon. Lett.*, vol. 30, no. 8, pp. 764–767, Aug. 2020.
[30] Q. A. Pham, D. Lopez-Bueno, T. Wang, G. Montoro, and P. L. Gilabert, "Multi-dimensional LUT-based digital predistorter for concurrent dual-band envelope tracking power amplifier linearization," in *Proc. IEEE Topical Conf. RF/Microw. Power Model. Radio Wireless Appl. (PAWR)*, Jan. 2018, pp. 47–50.
[31] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*, 2nd ed. São Paulo, Brazil: Krieger, 2014.
[32] C. Crespo-Cadenas, J. Reina-Tosina, M. J. Madero-Ayora, and J. Munoz-Cruzado, "A new approach to pruning volterra models for power amplifiers," *IEEE Trans. Signal Process.*, vol. 58, no. 4, pp. 2113–2120, Apr. 2010.
[33] O. Hammi, F. M. Ghannouchi, S. Boumaiza, and B. Vassilakis, "A data-based nested LUT model for RF power amplifiers exhibiting memory effects," *IEEE Microw. Wireless Compon. Lett.*, vol. 17, no. 10, pp. 712–714, Oct. 2007.
[34] Z. He, W. Ye, and S. Feng, "Digital predistortion based on multidimensional look-up table storing polynomial coefficients," *Electron. Lett.*, vol. 48, no. 22, pp. 1396–1397, Nov. 2012.
[35] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Cham, Switzerland: Springer, 2009.
[36] A. Brihuega *et al.*, "Piecewise digital predistortion for mmWave active antenna arrays: Algorithms and measurements," *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 9, pp. 4000–4017, Sep. 2020.
[37] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.
[38] Xilinx. *Performance and Resource Utilization for Complex Multiplier V6.0*. Accessed: Feb. 23, 2021. [Online]. Available: https://www.xilinx.com/html_docs/ip_docs/pru_files/cmpy.html
[39] Xilinx. *Performance and Resource Utilization for Adder/Subtracter V12.0*. Accessed: Feb. 23, 2021. [Online]. Available: https://www.xilinx.com/html_docs/ip_docs/pru_files/c-addsub.html
[40] K. Chapman. *Multiplexer Design Techniques for Datapath Performance With Minimized Routing Resources*. Accessed: Feb. 23, 2021. [Online]. Available: https://www.xilinx.com/support/documentation/application_notes/xapp522-mux-design-techniques.pdf
[41] A. Karalic, "Linear regression in regression tree leaves," in *Proc. ECAI*. Hoboken, NJ, USA: Wiley, pp. 440–441.
[42] D. S. Vogel, O. Asparouhov, and T. Scheffer, "Scalable look-ahead linear regression trees," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2007, pp. 757–764.
[43] J. C. Wang and T. Hastie, "Boosted varying-coefficient regression models for product demand prediction," *J. Comput. Graph. Statist.*, vol. 23, no. 2, pp. 361–382, Apr. 2014.
[44] Y. Shi, J. Li, and Z. Li, "Gradient boosting with piece-wise linear regression trees," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2018, pp. 3432–3438.
[45] A. S. Tehrani, H. Cao, S. Afsardoost, T. Eriksson, M. Isaksson, and C. Fager, "A comparative analysis of the complexity/accuracy tradeoff in power amplifier behavioral models," *IEEE Trans. Microw. Theory Techn.*, vol. 58, no. 6, pp. 1510–1520, Jun. 2010.
[46] 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.
[47] P. M. Tome, F. M. Barradas, T. R. Cunha, and J. C. Pedro, "Hybrid Analog/Digital linearization of GaN HEMT-based power amplifiers," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 1, pp. 288–294, Jan. 2019.
[48] N. Hammler, A. Cathelin, P. Cathelin, and B. Murmann, "A spectrum-sensing DPD feedback receiver with 30x reduction in ADC acquisition bandwidth and sample rate," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 9, pp. 3340–3351, Jun. 2019.
[49] Y. Li, X. Wang, and A. Zhu, "Sampling rate reduction for digital predistortion of broadband RF power amplifiers," *IEEE Trans. Microw. Theory Techn.*, vol. 68, no. 3, pp. 1054–1064, Mar. 2020.

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

He is currently a Post-Doctoral Researcher with the RF and Microwave Research Group, UCD. His current research interests include behavioral modeling and digital predistortion for radio frequency (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 pursuing the Ph.D. degree at University College Dublin (UCD), Dublin, Ireland.

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

Jingzhou Pang (Member, IEEE) received the B.S. degree in electrical engineering and the Ph.D. degree in circuits and systems from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2010 and 2016, respectively.

From December 2016 to July 2018, he was with Huawei Technologies Company Ltd., Chengdu, China, where he was an Engineer in charge of the research and development of 5G high-efficiency power amplifiers and transmitters. From July 2018 to August 2020, he was with the RF and Microwave Research Group, University College Dublin (UCD), Dublin, Ireland, where he was a Research Fellow in charge of the research of novel broadband transmitter architectures and radio frequency (RF)/microwave/mm-wave monolithic microwave integrated circuit (MMIC) power amplifiers. He is currently an Associate Professor with the School of Microelectronics and Communication Engineering, Chongqing University, Chongqing, China. His research interests include broadband high-efficiency power amplifier systems, bandwidth extension techniques for high-efficiency transmitters, and MMIC power amplifier design for RF/microwave and millimeter-wave applications.

Dr. Pang was a recipient of the EDGE Marie Sklodowska-Curie Individual Fellowship and the Third Place Award of the High Efficiency Power Amplifier Student Design Competition at the IEEE Microwave Theory and Techniques Society (IEEE MTT-S) International Microwave Symposium (IMS) in 2013.

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. He has published more than 150 peer-reviewed journal articles and conference papers. 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 MTT-S AdCom, the Chair of the Electronic Information Committee, and the Vice-Chair of the Publications Committee. 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 also the Chair of the MTT-S Microwave High-Power Techniques Committee. He has served as the Secretary of MTT-S AdCom in 2018. He is an Associate Editor of the *IEEE Microwave Magazine* and a Track Editor of the IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES.