On-Demand Real Time Optimizable Dynamic Model Sizing for Digital Predistortion of Broadband RF Power Amplifiers

Yue Li, *Student Member, IEEE,* and Anding Zhu, *Senior Member, IEEE*

***Abstract*—In this article, we present a dynamic model sizing approach for digital predistortion (DPD) of broadband radio frequency power amplifiers. By employing a novel model structure adaptation algorithm, the DPD model structure can be adaptively adjusted during its real time deployment to keep the optimum size and complexity under different operation conditions. Power consumption of DPD can be reduced by on-demand automatic model structure adaptation instead of reusing the same model structure for all power levels and band allocations. To realize dynamic model sizing, the adaptation algorithm explores new potential terms based on prior knowledge of the model structure, and prunes the DPD model with a stepwise backward regression method. Experimental results show that the algorithm can quickly find the optimum model structure when operation condition changes. During the adaptation, it can also maintain robust linearization performance with relatively low computational complexity and thus demonstrates itself as a suitable solution to the linearization of future broadband wireless systems.**

***Index Terms*—Behavioral modeling, digital predistortion, linearization, power amplifiers, pruning, Volterra series**

I. Introduction

In the next generation wireless communication systems, e.g., 5G, more new waveforms are expected to be adopted to improve spectral efficiency [1]. The resulting signals can lead to wider bandwidths and higher peak-to-average power ratio (PAPR) that may induce more complicated nonlinear behaviors in wireless transmitters, especially in the radio frequency (RF) power amplifiers (PAs). Digital predistortion (DPD) [2]–[4], a widely used linearization technique, is thereby faced with new challenges, because it may require a large model to compensate for the complex nonlinear distortion, which can lead to high power consumption in digital baseband [5].

In the past two decades, many DPD models have been proposed by pruning the full Volterra series, including memory polynomials (MP) [6], generalized memory polynomial (GMP) [7], dynamic deviation reduction (DDR) [8], and so on [9]. Recently, behavioral models employing piecewise structures have also shown great success, e.g., lookup tables (LUT) [10], splines [11], vector switched models [12], decomposed piecewise Volterra [13], decomposed vector rotation (DVR) [14] and magnitude-selective affine (MSA) [15]. Though these models achieve good performance and have substantially lower complexity than the full Volterra series, they may still require significant resources to comply with the linearity requirement posed by future wideband and multiband transmitters [16]. Moreover, as they define their basis functions based on prior knowledge on general PA behaviors [17], it is difficult to acutely adapt these models to individual PA characteristics and varying operation conditions.

The variations of PA behavior have been studied by directly accommodating such effects into behavioral models. A bandwidth and power scalable model was proposed in [18] which pre-specified different model sizes and pre-calculated corresponding LUT coefficients for each operation condition. Power-adaptive DPD in [19] linearized PAs under different power levels by dynamically updating DPD coefficients with a coefficient interpolation block. In [20], the authors adopted a similar strategy but updated DPD coefficients with a theoretically derived scaling rule instead of optimized parameters. A neural network model was developed in [21] which explicitly measures and models PA characteristics under predetermined frequency, voltage and temperature conditions. In [22], the long-term memory effects of gallium nitride (GaN) PAs are examined and compensated using physics-based models. To date, most solutions have been focusing on fast adaptation of the model coefficients, rather than optimization of model structures.

To find the optimum model structure for DPD and minimize its power consumption, a number of model pruning methods have been proposed to reduce the model size. In [23], Chen et al. developed an error variation ranking (EVR)-based method to select the basis functions that have highest impact on DPD performance. Hill-climbing method and genetic algorithm [24], [25] explores different model structures by local and global search of model hyperparameters. Particle swarm optimization (PSO) was adopted in [26] to search for suitable basis functions by optimizing an l_0-penalized cost function. Greedy matching pursuit methods, such as orthogonal matching pursuit (OMP) [27] and doubly orthogonal matching pursuit (DOMP) [28], were investigated in [29], which first ranks all available basis functions by matching pursuit algorithms and then selects the most important basis functions. In [30], the authors combined OMP search with simplified least squares (LS) model extraction to reduce the overall complexity of DPD systems. The adaptive basis function method in [31] proposed

Manuscript received Nov 18, 2019; revised Dec 30, 2019 and Jan 28, 2020; accepted Feb 18, 2020. This work was supported in part by the Science Foundation Ireland under Grant Numbers 13/RC/2077 and 17/NSFC/4850. (*Corresponding author: Yue Li*)

The authors are with the School of Electrical and Electronic Engineering, University College Dublin, Dublin, D04 V1W8, Ireland. (e-mail: yue.li1@ucdconnect.ie; anding.zhu@ucd.ie)

to adaptively add more terms into the model, but the DPD may be over-sized because the redundant basis functions could not be removed during DPD operation.

Despite numerous efforts in this field, the existing model pruning techniques still face challenges in real-time DPD deployment. One fundamental problem is that most approaches are designed for one-shot pruning, i.e., the model structure is fixed after the pruning process is completed. In realistic scenarios, signal characteristics and PA operation conditions can change significantly, and therefore the model structure obtained from one-shot pruning may perform badly or become over-sized under a different condition. To overcome this issue, it is desirable to optimize the model structure on the fly during the real time operation.

In this article, a novel method is proposed to adaptively update the model structure of DPD with low complexity and high robustness. With the ultimate goal of replacing the conventional LS-based coefficient update strategy, a dynamic model sizing approach is developed to update both model structure and coefficients on demand, which realizes adaptive pruning of DPD models for the first time. The core of the system, namely, a novel model structure adaptation algorithm, is proposed to satisfy the new challenges arising from this new application scenario. The algorithm works by adding in new terms based on prior knowledge of the model structure, and removing unimportant terms using a stepwise backward regression method. Unlike the existing approaches that focus on one-shot pruning, the proposed dynamic model sizing method maintains robust performance during the model adaptation process and achieves fast convergence with low complexity. It can thus optimize power consumption of DPD under varying operation conditions in real time.

The rest of this paper is organized as follows: Section II outlines the challenges in the deployment of existing pruning algorithms and presents a new system architecture with dynamic model sizing capability. In section III, a novel model structure adaptation algorithm suitable for real time deployment is detailed, featuring a statistical hypothesis-based model pruning method and a neighborhood-based model growing method. Section IV reports the experimental results, followed by a conclusion in Section V.

II. DPD Model Pruning

A. Deployment Issues of Model Pruning Algorithms

In conventional model pruning algorithms, the optimum DPD model is usually searched under a specific operation condition. Therefore, to deploy the DPD, we need to first find an optimum model structure through an off-line algorithm and implement the model in hardware. During DPD operation, the model structure is fixed and only the model coefficients are updated to track the variation of PA characteristics. A demonstration of the deployment procedures is illustrated in Fig. 1.

While many pruning algorithms have been demonstrated to successfully find small models with reasonably good linearization performance, inherent issues with the deployment strategy has prevented their application to realistic scenarios.

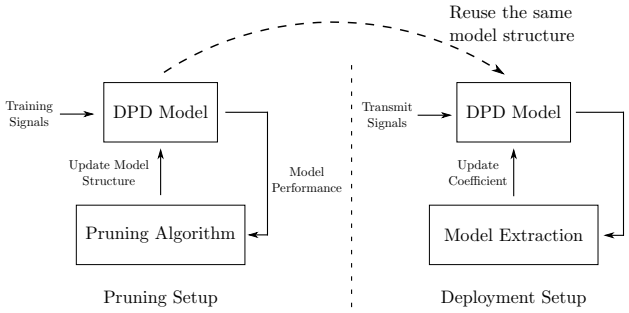

Fig. 1. Deployment procedures of conventional DPD model pruning algorithms.

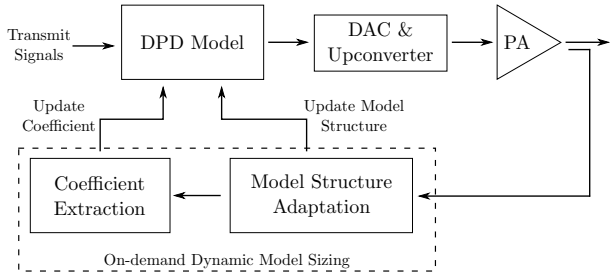

Fig. 2. System architecture of adaptive DPD model pruning.

A major concern is the robustness of these algorithms under varying operation conditions. As the model structure found by the pruning algorithms often works best only with the same test setup as that at the pruning stage, there is no guarantee that it can work well under different operation conditions. For example, a model searched with a 60 MHz signal may work poorly with 100 MHz signals, because the memory terms adopted by 60 MHz tests may be insufficient to compensate for the nonlinearity induced by 100 MHz signals.

One way to overcome this issue is to find a "worst-case" model that can work well in all scenarios. However, this approach introduces redundancy to the DPD model, which can lead to high complexity in both DPD actuator operation and model extraction process because of the large number of model coefficients. Another issue is that, some factors affecting PA characteristics may be hard to test, e.g., aging of electronic devices, making the model difficult to optimize. As a result, to tolerate the unseen factors, the final deployed model may be greatly over-sized.

B. Adaptive Pruning: Architecture and Challenges

In 5G systems, due to the use of small cells and beamforming technique, the PA power is significantly reduced compared to that in 4G systems and thus DPD power budget must be reduced too. To fully optimize DPD power consumption, it is desirable to employ optimized model structures while being resistant to varying operation conditions. To achieve this goal, we propose to update both the model structure and the model coefficients of DPD on the fly to cope with the variation of PA behavior, as shown in Fig. 2.

In this new architecture, the dynamic model sizing system can adaptively turn on/off specific hardware components to adjust the model size. In this way, only the required nonlinear

terms need to be activated and thus the minimum power consumption would be achieved. For instance, the hardware modules of a DPD implementation may include basis function generation and multiplication with model coefficients. The reduction of power consumption can be achieved by adaptively turning on/off these hardware components so that only a small fraction of hardware is activated at a time. Therefore, even though a redundant DPD model is still implemented in the DPD hardware, the average power consumption is expected to be much lower than that using conventional approaches.

Because of its adaptive nature, the new system architecture poses new requirements on model pruning that are difficult to satisfy with existing algorithms. Firstly, the algorithm must have a fast adaptation speed because the DPD needs to react acutely to the sudden changes of the PA behavior, such as those caused by power variations, and thus a fast model adaptation is required to quickly adjust the operation of DPD to maintain linearization performance. Secondly, we should use as few measurements as possible. A large number of measurements will not only increase complexity of the system operation but also increase power consumption because the feedback receiver must be activated many times. Thirdly, high linearization performance must be maintained during the adaptation period. Since the transmitter operates in real time, disruption to the system should be avoided. Therefore, if the pruning is applied during DPD operation, it is important to make sure all models being measured can produce good linearization performance so that the system linearity does not vary significantly during the adaptation. Finally, since the adaption algorithm usually needs to be run repeatedly, it is desirable to have low computational complexity. Otherwise, the overall system efficiency may deteriorate if the model pruning algorithm takes up too many resources.

III. Proposed Dynamic Model Sizing and Adaptation Algorithm

In this section, a novel model structure adaptation algorithm is proposed that can well address the challenges of adaptive model pruning problem while achieving good pruning performance. The algorithm starts from a given model structure and iteratively searches for a new model suitable for the current PA condition. To achieve desired objective, the algorithm explores new basis functions that are potentially beneficial for DPD modeling, and removes old ones that have negligible impact on linearization performance. Therefore, the update of structure mainly consists of two algorithmic steps, namely model pruning and model growing.

A. Model Pruning

The goal of the model pruning is to remove the unimportant terms in the model to make the model more efficient without degrading the performance. To do so, an effective and robust metric to measure the importance of model basis functions must be developed first.

In models that are linear in the parameters, the magnitude of coefficients are often interpreted as the contribution of the corresponding model term to the fitting problem. However, due to the strong correlations between different basis functions, severe multicollinearity exists in most DPD models. Therefore, a more robust approach is to test the statistical significance of the model coefficients [32].

A DPD model can be expressed as

$$\mathbf{y} = \mathbf{X}\mathbf{c} + \mathbf{e} \tag{1}$$

where $\mathbf{y} = [y(N), y(N-1), \cdots]^T$ is the desired output signal, $\mathbf{c}$ is a vector of model coefficients, and $\mathbf{e}$ is the modeling residue. $\mathbf{X}$ is the regression matrix containing all basis functions constructed with the input signal $\mathbf{x} = [x(N), x(N-1), \cdots]^T$, where N is the number of samples used and $(\cdot)^T$ represents transpose operation. For a MP model, for example, $\mathbf{X}$ can be represented as

$$\mathbf{X} = \left[\boldsymbol{\phi}_{10}^{\text{MP}}, \boldsymbol{\phi}_{20}^{\text{MP}}, \cdots, \boldsymbol{\phi}_{11}^{\text{MP}}, \boldsymbol{\phi}_{21}^{\text{MP}}, \cdots\right], \tag{2}$$

where

$$\boldsymbol{\phi}_{\boldsymbol{km}}^{\text{MP}} = \left[\phi_{km}^{\text{MP}}(N), \phi_{km}^{\text{MP}}(N-1), \cdots\right]^T \tag{3}$$

and

$$\phi_{km}^{\text{MP}}(n) = |x(n-m)|^{k-1}x(n-m). \tag{4}$$

The general framework in the context of model pruning is detailed in [29].

The coefficients $\mathbf{c}$ can usually be estimated by using LS as

$$\mathbf{c} = (\mathbf{X}^H\mathbf{X})^{-1}\mathbf{X}^H\mathbf{y}, \tag{5}$$

where $(\cdot)^H$ refers to Hermitian transpose. As $\mathbf{e}$ is orthogonal to all basis functions in the model, under large sample sizes, we assume $\mathbf{e}$ can be modeled as independent and identically distributed (i.i.d.) zero-mean Gaussian noise with variance σ^2. By viewing $\mathbf{X}$ as deterministic, $\mathbf{c}$ can be shown to obey a multi-variate Guassian distribution [32] whose covariance matrix can be derived as follows

$$\begin{aligned}\boldsymbol{\Sigma}_{\mathbf{c}} &= \mathbb{E}(\mathbf{c} - \mathbb{E}\mathbf{c})^2 \\ &= \mathbb{E}[(\mathbf{X}^H\mathbf{X})^{-1}\mathbf{X}^H(\mathbf{y} - \mathbb{E}\mathbf{y})]^2 \\ &= (\mathbf{X}^H\mathbf{X})^{-1}\mathbf{X}^H\mathbf{X}(\mathbf{X}^H\mathbf{X})^{-1}(\mathbf{y} - \mathbb{E}\mathbf{y})^2 \\ &= (\mathbf{X}^H\mathbf{X})^{-1}\sigma^2.\end{aligned} \tag{6}$$

Under this statistical modeling framework, the significance of model coefficients can be evaluated with hypothesis tests. To test the j-th coefficient c_j, we establish a null hypothesis that c_j has zero mean. Then we calculate the following z-score:

$$z_j = \frac{|c_j|}{\sigma\sqrt{v_j}} \tag{7}$$

where z_j is the z-score of c_j and v_j is the diagonal elements of matrix $(\mathbf{X}^H\mathbf{X})^{-1}$. All z_j's will form a vector $\mathbf{z}$.

Based on the statistical hypothesis test, the higher the z-score, the more likely this coefficient has a non-zero mean, which suggest that it should be kept in the model. Therefore, based on the z-score, the importance of all coefficients in the current model can be evaluated, and the term with the smallest z-score should be removed from the current model.

To avoid the high complexity of LS, an efficient pruning strategy is adopted. Consider a model with q coefficients, its regression matrix is $\mathbf{X}_{(\mathbf{q})}$, and its coefficient is $\mathbf{c}_{(\mathbf{q})}$. For convenience, we define $\mathbf{P}_{(\mathbf{q})} = ({\mathbf{X}_{(\mathbf{q})}}^H\mathbf{X}_{(\mathbf{q})})^{-1}$. Without loss

of generality, assume the last term in $\mathbf{X_{(q)}}$ is to be removed. Rewrite $\mathbf{P_{(q)}}$ into block matrix

$$\mathbf{P_{(q)}} = \begin{bmatrix} \mathbf{A} & \mathbf{u} \\ \mathbf{v}^T & d \end{bmatrix}, \tag{8}$$

where $\mathbf{A}$ is matrix, d is a scalar, and $\mathbf{u}$ and $\mathbf{v}$ are column vectors. Thus, $\mathbf{P_{(q)}}$ can be updated without matrix inversion,

$$\mathbf{P_{(q-1)}} = \mathbf{A} - \mathbf{u}\mathbf{v}^T/d. \tag{9}$$

Subsequently, $\mathbf{c_{(q)}}$ can be updated as

$$\mathbf{c_{(q-1)}} = \mathbf{P_{(q-1)}}\mathbf{X_{(q-1)}}\mathbf{y}. \tag{10}$$

The aforementioned method efficiently removes a basis function from the model. To remove multiple terms, the same algorithm is carried out iteratively.

By adopting the simplified computation procedures, it is not necessary to perform the full LS calculation, and the required computational complexity to update the model coefficients is very low. Firstly, the calculation of $(\mathbf{X_{(q-1)}}^H\mathbf{X_{(q-1)}})^{-1}$ can be achieved by (9), requiring only $(q-1)^2$ multiplications. Secondly, the calculation of $\mathbf{X_{(q-1)}}\mathbf{y}$ can be avoided by reusing the results of $\mathbf{X_{(q)}}\mathbf{y}$. Finally, the coefficients can be updated by multiplying the two quantities obtained from the previous steps, and the related complexity is also $(q-1)^2$. Therefore, after pruning the model, the coefficients of the new model can be efficiently re-estimated with low complexity.

B. Model Growing

Besides the pruning strategy, it is important to find potentially important terms that are not included in the current model. A straightforward approach is to consider all possible model terms in every iteration, but that will require high computational complexity and can reduce the robustness of model pruning due to a large number of coefficients involved. It is thus desirable to take into account only a subset of the full model terms that are considered useful in increasing model accuracy.

In this work, we take advantage of the z-scores calculated in the model pruning stage. It is believed that the important coefficients represent "good" directions in the feature space, which are worth further exploration. Let's take the MP model as an example. The nonlinear term $|x(n-m)|^{k-1}x(n-m)$ has polynomial order k and memory depth m, so it corresponds to the point (k, m) in the feature space. If it is shown to be important, we may infer that it well approximates some important nonlinear characteristics required by the DPD model. Nonlinear terms that lie in its neighborhood in the feature space, such as the terms corresponding to $(k \pm 1, m)$ and $(k, m \pm 1)$, are likely to provide similar modeling capability of these characteristics. Thus, new terms close to the important terms are added to the model during the model growing phase. A demonstration is shown in Fig. 3. Unlike the hill-climbing algorithm [25] which searches neighborhood in the hyperparameter space, e.g., perturbing polynomial order of the model, the proposed model growing method independently explores the neighborhood of every relevant model term, which naturally leads to sparse models.

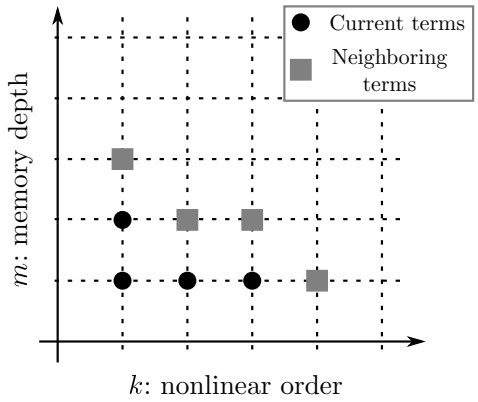

Fig. 3. Illustration of the model terms being added during model growing.

This strategy takes advantage of the prior knowledge of model structure, and applies to virtually all pruned Volterra models. Models, other than MP, can be similarly organized. For instance, basis functions in GMP model, i.e., $|x(n-m-p)|^{k-1}x(n-m)$, can be described as (k, m, p), and its neighborhood includes the terms located at $(k \pm 1, m, p)$, $(k, m \pm 1, p)$ and $(k, m, p \pm 1)$.

Different from existing model pruning algorithms which simply apply a sparsity constraint on the model structure, the neighborhood-based model growing method implicitly integrates general prior knowledge of behavioral modeling principles into the algorithm. By adding neighboring terms to the model, the algorithm is encouraged to explore model terms consecutive in the feature space. For example, the neighbors of $x(n-2)$ includes $x(n-1)$ and $x(n-3)$, and the three terms as a whole become a consecutive sequence in the dimension of memory depth. Thus, after model growing, we may have a local structure $\sum_{m=1}^{3} c_m x(n-m)$ in the model, which can be easily identified as a 3rd-order finite impulse response (FIR) filter. Since FIR filters have been widely used to approximate arbitrary linear frequency response, we believe these local structures have a better universal approximation ability than isolated model terms. Therefore, by employing the neighborhood-based model growing strategy, we can efficiently explore models having consecutive but irregular shape, such as the model structure shown in Fig. 3. In our opinion, models of this type are more likely to exhibit good linearization performance, which agrees with the experimental results presented in Fig. 11 of [23].

C. Full Algorithm

The dynamic model structure adaptation requires both model pruning and model growing to achieve satisfactory performance. Since the algorithm is targeted at real time DPD deployment, to reduce the number of data acquisitions and minimize disruption to the real time operation, in the proposed solution, we integrate the two processes into a full adaptation procedure with two nested loops. In the outer loop, input and output data from the PA are captured and the DPD structure and coefficients are updated at each iteration, while in the inner loop, iterative model pruning algorithm is conducted to remove redundant terms in the model. The detailed description

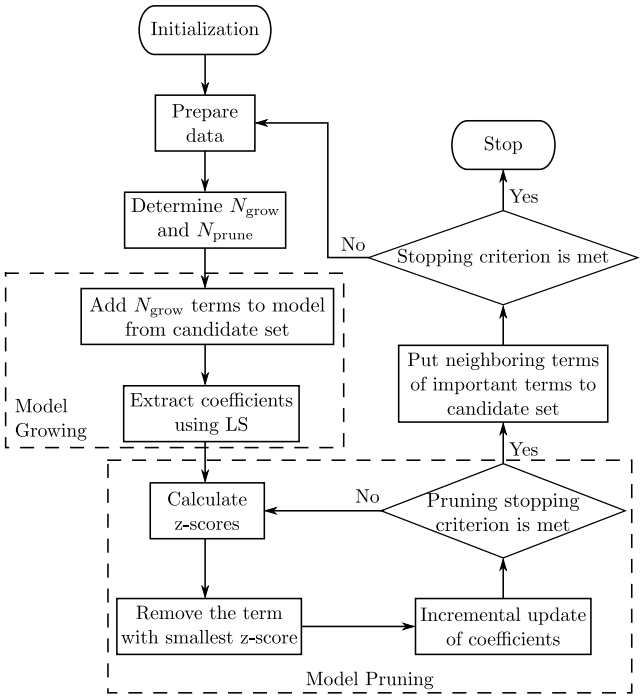

Fig. 4. Flow chart of dynamic model structure adaptation.

Algorithm 1 Dynamic Model Structure Adaptation

Input: Current model structure
Output: Updated model structure and coefficients

while stopping criterion is not met **do**
  Capture data and construct $\mathbf{X}$ and $\mathbf{y}$
  Determine N_{grow} and N_{prune} with N_v
  Add N_{grow} coefficients to the model from candidate set
  Calculate $\mathbf{c}$ using (5)
  for $t = 1$ to N_{prune} **do**
    Calculate $\mathbf{z}$ using (7)
    Find $r = \arg\min_j z_j$
    Remove c_r from $\mathbf{c}$
    Update $\mathbf{P}$ using (9)
    Update $\mathbf{c}$ using (10)
    if Performance degrades dramatically **then**
      Restore c_r to the model
      Break
    end if
  end for
  Build a candidate set with the neighboring terms of those with largest z-score
end while

of the full operation is given in Algorithm 1 and the flow of the procedures is illustrated in Fig. 4.

For initialization, to avoid starting from scratch, the model can start from a pre-determined model structure, such as a manually tuned MP or GMP model. At the start of each iteration, a new set of input and output data is captured and pre-processed by tasks, such as time alignment and model construction. System linearity is then evaluated to determine whether the model structure should be changed. The decision can be made by comparing the current linearity performance, typically specified by normalized mean squared error (NMSE) or adjacent channel power ratio (ACPR) values, with the desired target that defined by the specific application requirement. To keep the model size relatively stable, preventing explosion of the size, the maximum variation of model size in one iteration should be capped by a user-specified parameter N_v. That is to say, within one outer iteration, the model size would increase or decrease at most by N_v. N_v acts like a learning rate in the proposed algorithm: a smaller N_v can make the convergence slower while a larger N_v can lead to faster convergence, but the final performance may be affected if it is too large.

There are two cases in model adaptation process, depending on the performance of the initial model:

1) If the model performance does not reach the desired linearization requirement, the model needs to be expanded and more model terms should be added. In this case, the model growing algorithm is performed first, namely, N_{grow} nonlinear terms are added to the model from a candidate set, as described in Section III.B. The expanded model can then be extracted by using LS as in (5). Because the added new terms are not necessarily all effective and the new model must be optimized, the z-score based pruning process should be conducted to evaluate the impact of each term before finalizing the model structure, and the redundant terms are excluded in the final model. To limit the change in the number of coefficients to N_v at each iteration, we keep adding neighbors of important terms until the growing number N_{grow} reaches two times of N_v. Then in the pruning stage, the pruning number is set to $N_{\text{prune}} = N_{\text{grow}} - N_v$.
2) If the model is "too good", the model size should be reduced. It happens when DPD performance reaches an unnecessarily high level. We may also optionally reduce model size after the DPD performance first reaches desired target, so that the redundant terms added during model growing phase can be "dropped out". In this case, model growing is skipped, i.e., $N_{\text{grow}} = 0$, while N_{prune} ($\leq N_v$) coefficients are pruned in the model pruning stage.

As discussed earlier, the z-score based pruning procedure can be implemented in an efficient, iterative manner. Firstly, we calculate the z-score of all coefficients based on (7). The index of coefficients with the largest and smallest z-scores are recorded. Note that σ may be omitted in calculation since it does not affect ranking. Then the coefficients with smallest z-score is removed from the model. Backtracking technique can be optionally employed to monitor performance variation. Once modeling error is found to degrade significantly, the deleted coefficient can be restored to the model. In practice, we can first calculate the difference between current and target NMSE values, and then force the maximum NMSE degradation caused by pruning to be smaller than this margin. Then the coefficients will be updated using (9) and (10). The procedures of z-score calculation, basis function removal and coefficient update are iterated until all of the N_{prune} coefficients

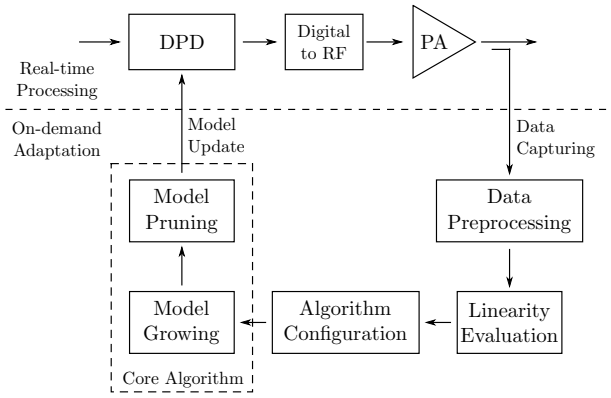

Fig. 5. Deployment of dynamic model sizing.

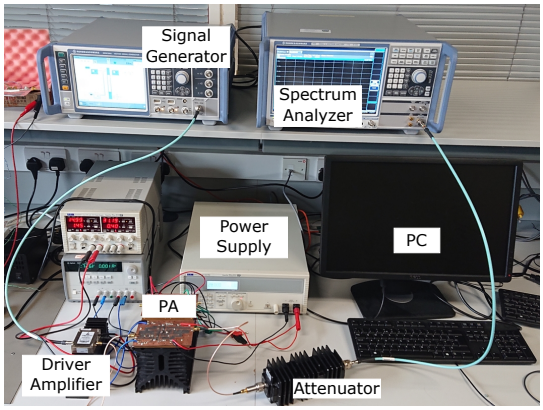

Fig. 6. The photograph of the DPD test bench.

are removed or backtracking takes effect. Finally, the terms with largest z-scores are located, and their neighboring terms in the feature space will form a new candidate set for use in the next update of model structure.

D. Real-time Deployment

The proposed algorithm can be easily deployed in a real-time DPD system. As shown in Fig 5, the resulting DPD system can be divided into two parts: the DPD module that processes transmit data in real time, and the model adaptation unit that operates on-demand. The adaptation module is triggered when PA behavior is known to change and stops after the adaptation reaches convergence.

To make DPD adaptable to a wide range of scenarios, a large set of candidate model terms may be implemented in digital unit. However, in real-time operation, not all the hardware components are operational and they can be turned on/off by different control methods depending on the hardware platform and configuration. For example, a basic strategy is to add a control bit to every coefficient and use that bit to do clock or power gating for the coefficient multipliers.

When an update of DPD is desired, new data are captured from PA output and the model structure adaptation algorithm will operate. The process starts from data preprocessing, including time alignment and model construction. The algorithm parameters are then determined based on the linearity of captured signals. The core algorithmic steps, model growing and model pruning, are then performed to obtain the new model structure and coefficients, which are finally loaded to the DPD module in the real-time processing path. It is worth mentioning that data capturing and model update are performed only once in the outer loop iteration of the adaptation algorithm. The internal processing of the algorithm, e.g., the iterative pruning steps in the inner loop, will not interact with the main DPD module. In other words, the DPD update will take effect only after all internal steps are finished. By employing the proposed approach, the number of data acquisition is minimized and disruption to the real time operation can be avoided.

IV. Results

A. Experimental Results

To validate the proposed method, it was experimentally compared against other approaches. Thus, a test platform was set up, as shown in Fig. 6, which included PC, signal generator, driver amplifier, PA, attenuator and spectrum analyzer. The PA under test was an in-house designed broadband Gallium Nitride (GaN) Doherty power amplifier operating at 3.1/3.3 GHz with the maximum output power at 43 dBm. The excitation input signals were carrier-aggregated LTE signals with 6.5 dB PAPR. The GMP model was employed in DPD. Recorded I/Q input and output samples were time aligned and normalized before training the model. The model extraction was based on indirect learning and performed in MATLAB.

To emulate the variations of PA characteristics in real environment, the PA was measured under five different test cases where each case corresponds to a different operation condition:

I) 100 MHz modulated signal at 3.1 GHz carrier frequency with 35 dBm average output power;
II) 20 MHz modulated signal at 3.1 GHz carrier frequency with 35 dBm average output power;
III) 80 MHz modulated signal at 3.3 GHz carrier frequency with 32 dBm average output power;
IV) 80 MHz modulated signal at 3.3 GHz carrier frequency with 35 dBm average output power;
V) 20+20 MHz non-contiguous modulated signal (60 MHz instantaneous bandwidth) at 3.3 GHz carrier frequency with 35 dBm output power.

It can be observed that Case II has reduced bandwidth, while Case III decreased power level and changed the carrier frequency. In Case IV, the power level was increased. In Case V, the test signal was changed to a 40 MHz non-contiguous carrier aggregated LTE signal. Thus, the PA was expected to behave differently in these cases.

In the experimental tests, the proposed method was tested sequentially on the five cases, namely, the model in the final state of the earlier test was used as the initial model of the following test. In other words, the final model used in the test I was used as the starting model in the test II, and so on. The sampling rate of the baseband signals was set

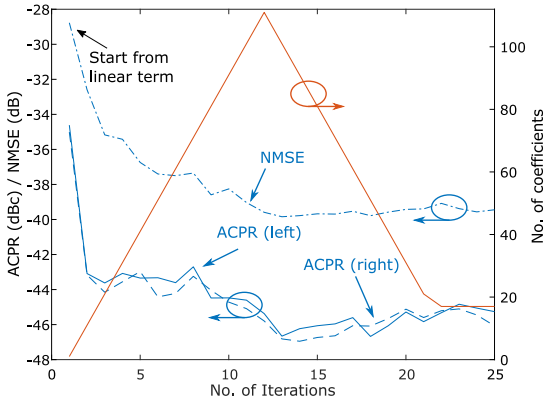

Fig. 7. Performance and model size during iterations in Case I.

to 5 times the signal bandwidth in each case. In all cases, 20,000 data samples were used in DPD model extraction, and each measurement used different test signals. To verify the performance of pruning algorithm, manual tuning and and a latest pruning technique, DOMP [28], were employed for comparison. The two methods were tested on the five cases independently to show the performance of pruning algorithm itself without considering deployment issues. We also compared different deployment strategies by testing a "fixed DPD" method which used the same model structure in all test cases. The common model structure was obtained by manual tuning from the most challenging case, i.e., Case I.

The manual tuning method followed a similar strategy as the hill-climbing method in [24]. At a given model structure, we tested its neighboring models by increasing or decreasing each of the parameters by one (nonlinear order, memory depth and leading/lagging depth), and chose the best model among them. Based on the new model structure, we repeated the search again, until there was no performance improvement. In the test of DOMP, a large GMP model with 322 coefficients were employed in the DOMP algorithm. To reduce computational complexity, we stopped the DOMP ranking procedure after the model size or NMSE performance reached that of the tuned DPD. After the ranking, the model with the best Bayesian information criterion was selected. In the proposed method, the parameter N_v was set to 10. Also, we considered the model to be "too good" if the achieved NMSE exceeded target NMSE by more than 0.4 dB.

1) Case I: The first test case used 100 MHz signal at 3.1 GHz and produced 35 dBm average output power. As is shown in Fig. 7, in the 100 MHz test, we started from a basic model and gradually increase the model size. The algorithm explored larger models in the first few iterations to include all necessary terms, before reducing the model size to reach the optimum solution after 22 iterations. The proposed method achieved comparable performance with the manually tuned model while having a much smaller size. A summary of test results is given in Table I and the spectra of PA output are shown in Fig. 8.

2) Case II: The second condition had the following settings: 20 MHz signal, 3.1 GHz carrier frequency and 35 dBm output power. The algorithm was initialized with the last model

TABLE I
PERFORMANCE COMPARISON IN CASE I

	No. of Coefficients	NMSE (dB)	ACPR (dBc) (lower/upper band)
w/o DPD	N/A	-28.8	-35.2/-34.4
Tuned DPD	133	-39.5	-44.9/-45.0
DOMP	84	-36.8	-43.0/-43.4
Proposed	17	-39.4	-46.1/-45.3

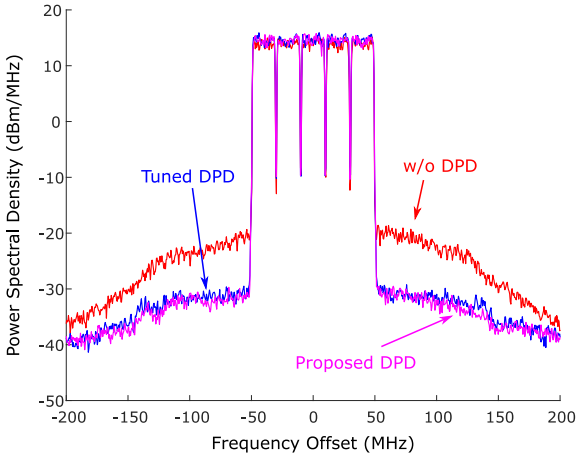

Fig. 8. Output spectrum comparison for Case I.

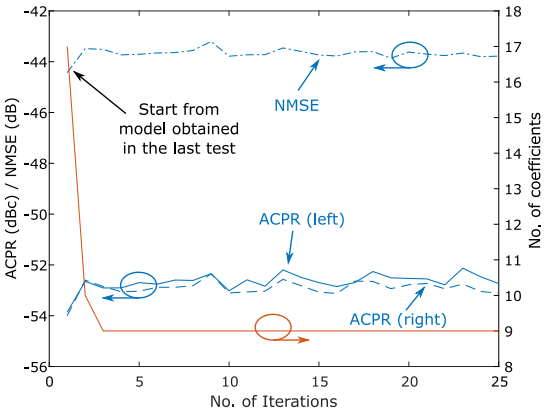

Fig. 9. Performance and model size during iterations in Case II.

TABLE II
PERFORMANCE COMPARISON IN CASE II

	No. of Coefficients	NMSE (dB)	ACPR (dBc) (lower/upper band)
w/o DPD	N/A	-30.3	-37.3/-38.2
Fixed DPD	133	-43.2	-51.7/-52.2
Tuned DPD	64	-43.5	-52.1/-52.7
DOMP	54	-43.4	-51.0/-51.8
Proposed	9	-43.8	-53.1/-52.7

obtained from the 100 MHz test. As the nonlinearity was weaker, the model size was decreasing during the adaptation. From Fig. 9, the optimum model was obtained after 3 iterations. In this case, the proposed method achieved the

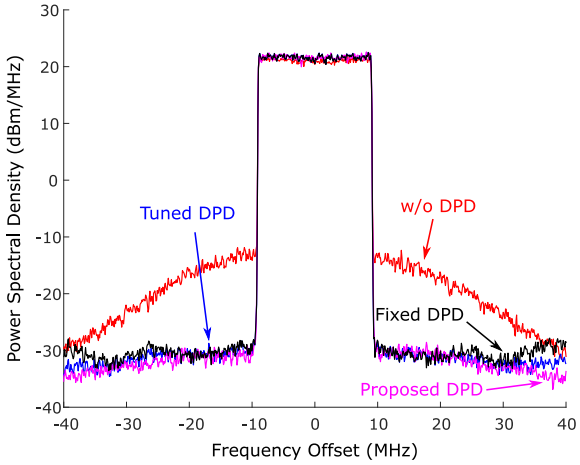

Fig. 10. Output spectrum comparison for Case II.

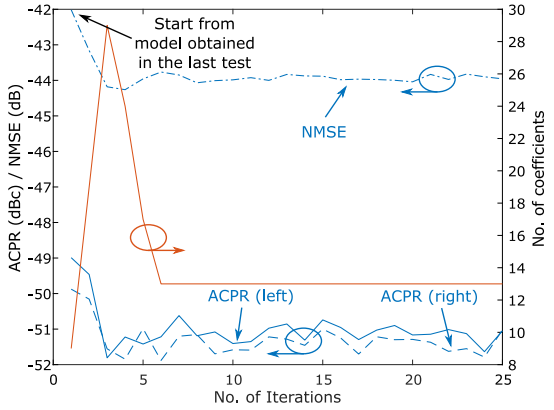

Fig. 11. Performance and model size during iterations in Case III.

smallest model size with similar linearization performance as that of other approaches. It also shows that the fixed DPD has almost identical performance as the tuned one and the proposed method, but has a larger size. The performance of different methods is summarized in Table II and the spectrum results are shown in Fig. 10.

3) Case III: In this test, the signal bandwidth, power level and carrier frequency were changed to 80 MHz, 32 dBm and 3.3 GHz, respectively. The PA was then operated at approximately 4 dB back off from maximum power. The algorithm started with the optimum model from the 20 MHz test. Since all parameters were changed, the PA behavior varied accordingly. As shown in Fig. 11, the algorithm converged after 6 iterations. The proposed method again acquired the smallest model with reasonable linearization performance. The performance summary and spectrum plots are shown in Table III and Fig. 12, respectively.

4) Case IV: In the fourth case, the output power was increased to 35 dBm, and other settings were kept unchanged (80 MHz signal @ 3.3 GHz). The algorithm continued from the last test with a 3 dB higher power level. As shown in Fig. 13, the algorithm converged after 6 iterations. A summary of linearization performance is shown in Table IV and the spectrum results are plotted in Fig. 14.

TABLE III
PERFORMANCE COMPARISON IN CASE III

	No. of Coefficients	NMSE (dB)	ACPR (dBc) (lower/upper band)
w/o DPD	N/A	-28.6	-35.3/-34.8
Fixed DPD	133	-43.7	-51.5/-52.0
Tuned DPD	80	-44.3	-51.9/-52.1
DOMP	54	-44.8	-52.3/-52.5
Proposed	13	-44.0	-51.0/-51.0

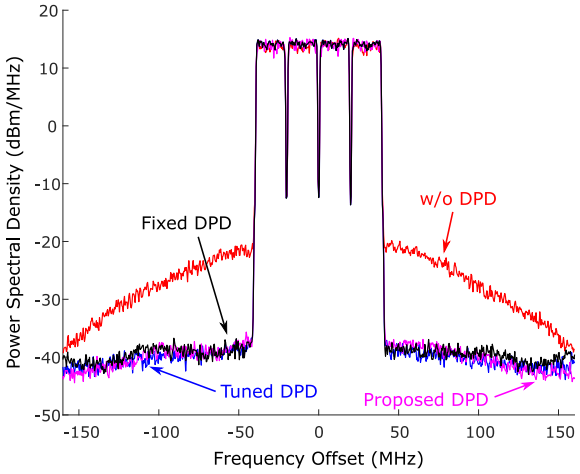

Fig. 12. Output spectrum comparison for Case III.

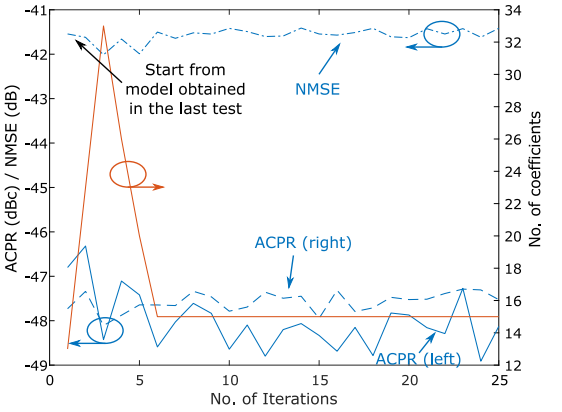

Fig. 13. Performance and model size during iterations in Case IV.

TABLE IV
PERFORMANCE COMPARISON IN CASE IV

	No. of Coefficients	NMSE (dB)	ACPR (dBc) (lower/upper band)
w/o DPD	N/A	-28.1	-34.7/-35.7
Fixed DPD	133	-41.7	-48.3/-48.7
Tuned DPD	95	-41.8	-48.0/-48.5
DOMP	93	-41.0	-46.3/-47.1
Proposed	15	-41.4	-47.5/-48.1

5) Case V: In the last case, the test signal was changed to a non-contiguous carrier-aggregated LTE signal with the same

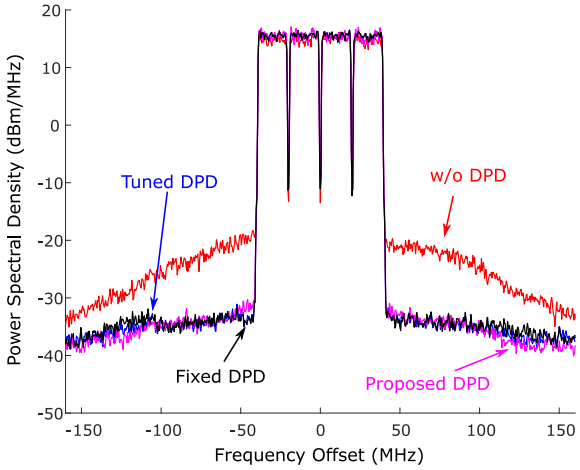

Fig. 14. Output spectrum comparison for Case IV.

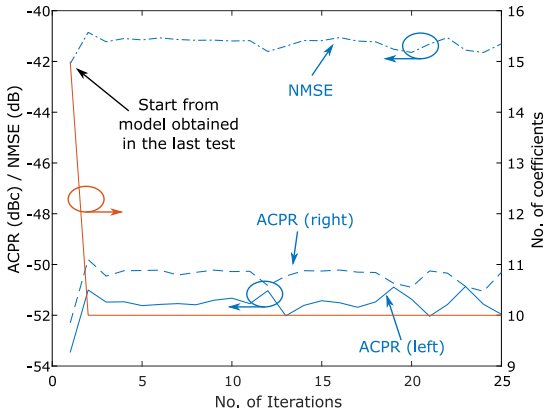

Fig. 15. Performance and model size during iterations in Case V.

TABLE V
PERFORMANCE COMPARISON IN CASE V

	No. of Coefficients	NMSE (dB)	ACPR (dBc) (lower/upper band)
w/o DPD	N/A	-28.4	-37.5/-37.3
Fixed DPD	133	-40.8	-50.1/-50.1
Tuned DPD	76	-41.2	-51.5/-51.3
DOMP	73	-37.3	-43.8/-45.1
Proposed	10	-41.3	-50.3/-52.0

frequency/power setting. As shown in Fig. 15, the algorithm converged after 2 iterations. The corresponding linearization performance is shown in Table V and the spectrum results are illustrated in Fig. 16.

To sum up, the experimental results show that the proposed algorithm maintains robust performance during the model structure adaptation. Compared to the other approaches, the proposed algorithm achieves comparable performance with a smaller model size under all operation conditions.

B. Complexity Analysis

In this part, the power consumption and computational complexity of the proposed DPD are analyzed. We consider

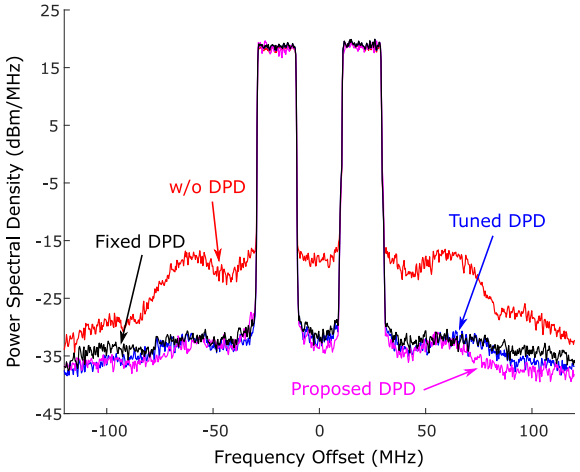

Fig. 16. Output spectrum comparison for Case V.

the same dynamic environment as the previous part.

In the DPD actuator part, the fixed DPD approach uses the conventional model sizing strategy and needs to use a worst-case (largest) model all the time. It means that, in Case II, the DPD model being deployed still had 133 coefficients, though 9 coefficients would be enough with the proposed method. On the contrary, in the proposed approach, the size of the DPD model can be dynamically optimized with the dynamic model sizing system. It not only can adapt the system to different operation conditions, but also find an optimized sparse model structure for each case. The proposed method therefore is expected to achieve lower overall power consumption than the conventional fixed DPD approach.

To compare the power consumption of DPDs, we analyze the complexity of activated hardware components in each case. According to [9], most hardware complexity of GMP model comes from the coefficient multiplication. Thus, we use the complexity of coefficient multiplication as a primary indicator of power consumption. In the analysis, each complex multiplier is assumed to be composed of 3 real multipliers.

The comparison is presented in Table VI. The proposed method activated significantly fewer hardware components than the conventional fixed DPD approach in all test cases, which leads to roughly 87-93% reduction in dynamic power consumption.

The algorithmic complexity of the pruning algorithm is also analyzed, by comparing it to the latest greedy matching pursuit method DOMP. To calculate the per-iteration complexity, we assume there are q terms in the current model and Q terms in the entire model, and we use N data samples in calculation. The complexity comparison is shown in Table VII, which suggests both algorithms have similar computational complexity per iteration, while the proposed method converged faster. In our opinion, it is because DOMP needs to rank all coefficients, so the required number of iterations is closely related to the total model size. In the proposed algorithm, however, the convergence speed mainly depends on the variation of PA characteristics. If the PA behavior does not change significantly, the algorithm can converge very fast, which has been confirmed in Fig. 9 and 15.

TABLE VI
COMPARISON OF DIFFERENT DEPLOYMENT STRATEGIES

		Fixed DPD w/ Conventional Model Sizing	Proposed DPD
Case I: 100 MHz BW 35 dBm@3.1 GHz	No. of Coefficients	133	17
	No. of Active Multipliers	399	51
Case II: 20 MHz BW 35 dBm@3.1 GHz	No. of Coefficients	133*	9
	No. of Active Multipliers	399*	27
Case III: 80 MHz BW 32 dBm@3.3 GHz	No. of Coefficients	133*	13
	No. of Active Multipliers	399*	39
Case IV: 80 MHz BW 35 dBm@3.3 GHz	No. of Coefficients	133*	15
	No. of Active Multipliers	399*	45
Case V: 20 + 20 MHz BW 35 dBm@3.3 GHz	No. of Coefficients	133*	10
	No. of Active Multipliers	399*	30

*An over-sized model is required when there is no model structure adaptation.

TABLE VII
ALGORITHMIC COMPLEXITY COMPARISON OF DIFFERENT PRUNING ALGORITHMS

	DOMP [28]	Proposed
Per-iteration Complexity	$\mathcal{O}(q^2N + QN)$	$\mathcal{O}(q^2N + q^2N_{\text{prune}})$
No. of Iterations	$\mathcal{O}(Q)$	< 20

V. CONCLUSION

In this paper, we have proposed a novel dynamic model sizing method that realizes adaptive model pruning for real-time DPD systems. The proposed approach well satisfies all the requirements put forward by the adaptive pruning problem.

Firstly, the model growing and pruning steps can efficiently find promising model structure by extracting useful information from the current model. Since it does not have to start from scratch, it can converge within a few iterations. Secondly, the model being loaded and measured has a robust structure produced by growing and pruning steps, which has been experimentally verified to achieve stable performance. Additional control imposed by the capping of model size variation and the optional backtracking technique provides further guarantee on the model performance in the real-time environment. Finally, the proposed method also has low computational complexity and thus is well suitable for deployment in future communication systems.

REFERENCES

[1] M. Shafi, A. F. Molisch, P. J. Smith, T. Haustein, P. Zhu, P. D. Silva, F. Tufvesson, A. Benjebbour, and G. Wunder, "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*. Artech House, 2014.

[3] F.-L. Luo, *Digital Front-End in Wireless Communications and Broadcasting: Circuits and Signal Processing*. Cambridge, UK: Cambridge University Press, 2011.

[4] 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.

[5] 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.

[6] L. Ding, G. Zhou, D. Morgan, Z. Ma, J. Kenney, J. Kim, and C. Giardina, "A robust digital baseband predistorter constructed using memory polynomials," en, *IEEE Trans. Commun.*, vol. 52, no. 1, pp. 159–165, Jan. 2004.

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

[8] 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.

[9] 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.

[10] 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.

[11] N. Naraharisetti, P. Roblin, C. Quindroit, and S. Gheitanchi, "Efficient least-squares 2-D-cubic spline for concurrent dual-band systems," *IEEE Trans. Microw. Theory Techn.*, vol. 63, no. 7, pp. 2199–2210, Jul. 2015.

[12] 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, 2012.

[13] A. Zhu, P. Draxler, C. Hsia, T. Brazil, D. Kimball, and P. Asbeck, "Digital predistortion for envelope-tracking power amplifiers using decomposed piecewise Volterra series," *IEEE Trans. Microw. Theory Techn.*, vol. 56, no. 10, pp. 2237–2247, Oct. 2008.

[14] 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.

[15] 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.

[16] P. L. Gilabert, G. Montoro, D. Vegas, N. Ruiz, and J. A. Garcia, "Digital predistorters go multidimensional: DPD for concurrent multiband envelope tracking and outphasing power amplifiers," *IEEE Microw. Mag.*, vol. 20, no. 5, pp. 50–61, May 2019.

[17] A. Zhu, J. C. Pedro, and T. R. Cunha, "Pruning the Volterra series for behavioral modeling of power amplifiers using physical knowledge," *IEEE Trans. Microw. Theory Techn.*, vol. 55, no. 5, pp. 813–821, May 2007.

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

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

[20] C. Crespo-Cadenas, M. J. Madero-Ayora, J. Reina-Tosina, and J. A. Becerra-González, "Transmitter linearization adaptable to power-varying operation," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 10, pp. 3624–3632, Oct. 2017.

[21] R. Gracia and N. Medrano-Marques, "RF power amplifier linearization in professional mobile radio communications using artificial neural networks," *IEEE Trans. Ind. Electron.*, vol. 66, no. 4, pp. 3060–3070, Apr. 2019.

[22] 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.

[23] W. Chen, S. Zhang, Y.-J. Liu, F. M. Ghannouchi, Z. Feng, and Y. Liu, "Efficient pruning technique of memory polynomial models suitable for PA behavioral modeling and digital predistortion," *IEEE Trans. Microw. Theory Techn.*, vol. 62, no. 10, pp. 2290–2299, Oct. 2014.

[24] S. Wang, M. A. Hussein, O. Venard, and G. Baudoin, "A novel algorithm for determining the structure of digital predistortion models," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7326–7340, Aug. 2018.

[25] S. Wang, M. A. Hussein, G. Baudoin, O. Venard, and T. Gotthans, "Comparison of hill-climbing and genetic algorithms for digital predistortion models sizing," in *2016 IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, Dec. 2016, pp. 289–292.

[26] A. H. Abdelhafiz, O. Hammi, A. Zerguine, A. T. Al-Awami, and F. M. Ghannouchi, "A PSO based memory polynomial predistorter with embedded dimension estimation," *IEEE Trans. Broadcast.*, vol. 59, no. 4, pp. 665–673, Dec. 2013.

[27] J. Reina-Tosina, M. Allegue-Martínez, C. Crespo-Cadenas, C. Yu, and S. Cruces, "Behavioral modeling and predistortion of power amplifiers under sparsity hypothesis," *IEEE Trans. Microw. Theory Techn.*, vol. 63, no. 2, pp. 745–753, Feb. 2015.

[28] J. A. Becerra, M. J. Madero-Ayora, J. Reina-Tosina, C. Crespo-Cadenas, J. García-Frías, and G. Arce, "A doubly orthogonal matching pursuit algorithm for sparse predistortion of power amplifiers," *IEEE Microw. Compon. Lett.*, vol. 28, no. 8, pp. 726–728, Aug. 2018.

[29] J. A. Becerra, M. J. Madero-Ayora, and C. Crespo-Cadenas, "Comparative analysis of greedy pursuits for the order reduction of wideband digital predistorters," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 9, pp. 3575–3585, Sep. 2019.

[30] Q. A. Pham, G. Montoro, D. López-Bueno, and P. L. Gilabert, "Dynamic selection and estimation of the digital predistorter parameters for power amplifier linearization," *IEEE Trans. Microw. Theory Techn.*, vol. 67, no. 10, pp. 3996–4004, Oct. 2019.

[31] X. Yu and H. Jiang, "Digital predistortion using adaptive basis functions," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 12, pp. 3317–3327, Dec. 2013.

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