Cascaded Spline-Based Models for Complex Nonlinear Systems: Methods and Applications

Pablo Pascual Campo, *Graduate Student Member, IEEE*, Lauri Anttila, *Member, IEEE*, Dani Korpi, and Mikko Valkama, *Senior Member, IEEE*

***Abstract*—In this paper, we present a class of cascaded nonlinear models for complex-valued system identification, aimed at baseband modeling of nonlinear radio systems. The proposed models consist of serially connected elementary linear and nonlinear blocks, with the nonlinear blocks implemented as uniform spline-interpolated look-up tables (LUT) and the linear blocks as FIR filters. Wiener, Hammerstein, and Wiener-Hammerstein models are built, and simple but efficient gradient based adaptive learning rules are derived for all the models. This approach leads to remarkably simple solutions in terms of computational complexity, making the techniques suitable for real-time implementation. The proposed methods are then applied to full-duplex self-interference cancellation and digital predistortion in various real-life scenarios. First, evaluations with measured data from an in-band full-duplex prototype working at 2.4 GHz ISM band show that the algorithms are capable of obtaining similar cancellation performance as existing state-of-the-art solutions, regardless of the clearly reduced complexity. Second, a mmW active antenna array working at 28 GHz center frequency is digitally predistorted with the proposed solutions. The unwanted emissions and nonlinear distortion are suppressed to similar levels as with other state-of-the art solutions, and the corresponding linearity specifications are fulfilled in all cases, while the processing complexity is again drastically reduced.**

***Index Terms*—Behavioral modeling, cascaded models, digital predistortion, full-duplex, linearization, look-up tables, power amplifiers, self-interference cancellation, splines.**

I. Introduction

NONLINEAR modeling and system identification are important ingredients in many wireless communications systems, due to the inherent nonlinearity of certain hardware components. The power amplifier (PA), for example, is typically operated in its nonlinear region to improve power efficiency [1]. Behavioral modeling and digital predistortion (DPD) of the PA are typical examples of nonlinear modeling tasks [2]. Other applications include, for example, digital self-interference cancellation (DSIC) in simultaneous transmit and receive (STAR) devices [3]–[5] and nonlinear channel equalization in satellite communications [6], [7].

Most often a general black-box or behavioral modeling approach is adopted for such problems, for example based on the Volterra series [8], [9] or artificial neural networks [7], [10]. However, in many cases some physical knowledge of the system under study is available, and a block-oriented model is well argued [7]. A good example of such a problem is self-interference cancellation in STAR or in-band full-duplex (IBFD), where the modeling is typically concerned with identifying a nonlinear device (the PA) in cascade with a linear system (the self-interference propagation channel), which together constitute a Hammerstein system [11]. Nonetheless, cascaded models have been successfully applied to black-box modeling problems as well such as DPD, see for example [12]–[15]. Compared to linear-in-parameters models such as Volterra, cascaded models typically have much less free parameters, thus they are appealing when low-complexity implementation is pursued.

The literature is abundant in complex nonlinear models and learning techniques for PA behavioral modeling, DPD, and DSIC. The vast majority of the existing works are based on Volterra-type models; see [2], [16] for overviews of these approaches. The literature on cascaded models and the associated learning methods for communications applications is more scarce. Techniques in this category contain the works in [11]–[15], [17]–[22]. In [20], [21], a frequency-domain identification of Hammerstein and Wiener-Hammerstein DPD models, respectively, was proposed. These works relied on the direct learning approach, i.e., required learning the PA forward model which was then inverted. In [12], a Hammerstein DPD with polynomial nonlinearity was proposed, together with an offline two-stage least-squares (LS) estimation scheme based on the indirect learning architecture (ILA). In [18], the authors proposed a Hammerstein DPD with a look-up table (LUT) nonlinearity. They also resorted to a two-stage estimation procedure and ILA, utilizing smoothed AM-AM and AM-PM curves for generating the LUT, and LS for estimating the filter coefficients. The study in [14] proposed a parallel Hammerstein (PH) DPD, with two independent real-valued spline-based Hammerstein models for the magnitude and phase. ILA with the Levenberg-Marquardt algorithm was used for parameter estimation, making real-time learning and tracking challenging due to the involved computational complexity. In [17] and [15], the authors presented Wiener and Hammerstein models, respectively, building on spline-based

Manuscript received June 23, 2020; revised November 11, 2020; accepted December 12, 2020. Date of publication December 23, 2020; date of current version January 14, 2021. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. A. A. Rontogiannis. This work was supported in part by the Academy of Finland under the projects 301820, 323461, 332361, and 319994. *(Corresponding author: Pablo Pascual Campo.)*

Pablo Pascual Campo, Lauri Anttila, and Mikko Valkama are with the Department of Electrical Engineering, Tampere University, 33720 Tampere, Finland (e-mail: pablo.pascualcampo@tuni.fi; lauri.anttila@tuni.fi; mikko.valkama@tuni.fi).

Dani Korpi is with Nokia Bell Labs, 02610 Espoo, Finland (e-mail: dani.korpi@nokia-bell-labs.com).

Digital Object Identifier 10.1109/TSP.2020.3046355

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

neural networks. The learning is based on the direct learning architecture, where the forward model is estimated with LS based algorithms, and then inverted with an iterative scheme. Altogether, the learning complexities of all the above techniques are considerable, and can be considered generally unsuitable for real-time implementation. In [22], a Hammerstein DPD based on the simplical canonical piece-wise linear (SCPWL) basis functions was introduced, and least mean squares (LMS)-based learning algorithms with both direct and indirect learning architecture were proposed. However, the solutions required the identification of the forward PA model, which was assumed to obey the Wiener model, thus complicating the estimation procedure. The Hammerstein DPD model was also formulated into a linear-in-parameters form (effectively a PH model), which led to overparameterization.

In the context of DSIC, PH based algorithms building on polynomial nonlinearities were proposed in [23]–[26], with [23]–[25] operating in time-domain and [26] in the frequency-domain. The works in [23], [24] utilized block LS estimation, while adaptive estimation methods were used in [25], [26]. While the PH model has been shown to be an accurate model for DSIC, it suffers from high complexity, making real-time implementation very challenging, especially with wider bandwidths. In [11], [27], which serve as a starting point or background for this paper, we proposed a Hammerstein model for DSIC and DPD, respectively, building on spline-interpolated LUTs and simple adaptive learning rules.

In this paper, we propose block-oriented models for complex-valued baseband modeling of nonlinear radio systems, along with simple gradient-based learning rules. We build Wiener, Hammerstein, and Wiener-Hammerstein models, comprised of different combinations of spline-interpolated LUTs and finite impulse response (FIR) filters. The techniques are based on the recently introduced concept of spline adaptive filters, which was developed for real-valued systems in [28]–[30]. The proposed techniques differ from these works in the following ways:

- The adaptive spline interpolation scheme within the cascaded models is specifically developed for complex-valued modeling of radio systems, instead of real-valued systems;
- We introduce a so-called injection-based nonlinear block, where the complex nonlinear gain of the system is defined as a deviation from unity gain. This helps to control the gain ambiguity between the cascaded blocks, and also reduces the dynamic range and thus the number of bits of the LUT;
- We point out and fix a shortcoming in the original learning equations in [28]–[30], which slowed down their convergence;
- We apply the techniques to two real-life problems: self-interference cancellation in IBFD, and digital predistortion of nonlinear power amplifiers;
- We verify the functionality and performance of the models using measured signals, and compare to the widely-used memory polynomial (MP) and generalized memory polynomial (GMP).

The proposed techniques are shown to offer similar performance compared to the MP and GMP models in both direct and inverse modeling problems using measured signals, but with greatly reduced processing complexity. Altogether, the framework proposed in this paper offers appealing low-complexity adaptive solutions for real-time applications that require complex-valued nonlinear model identification. Thus the paper offers contributions in both signal processing theory and applications.

The rest of the paper is organized as follows. First, Section II presents the general real-valued theory of spline interpolation, and then extends it to the complex domain. Building on our early work in [11], Section III then presents the spline-interpolated Hammerstein model, provides a complexity analysis, and highlights the differences to the original real-valued work in [29]. Section IV and Section V describe the spline-interpolated Wiener and Wiener-Hammerstein approaches, along with their learning rule derivations and complexity analyses. In Section VI, we present two applications for the techniques, along with experiments in real-life use cases, in order to verify and validate the proposed models. Finally, Section VII summarizes the main findings of this paper.

I. Notation used in this paper

In this paper, matrices are represented by capital boldface letters, e.g., $\mathbf{\Sigma} \in \mathbb{C}^{M\times N}$. Ordinary transpose, Hermitian transpose, and complex conjugation are denoted by $(\cdot)^T$, $(\cdot)^H$, and $(\cdot)^*$, respectively. By default, vectors are complex-valued column vectors, presented with lowercase boldface letters, i.e., $\mathbf{v} \in \mathbb{C}^{M\times 1} = [v_1\, v_2\, \cdots\, v_M]^T$. Additionally, the absolute value, floor, and ceil operators are represented as $|\cdot|$, $\lfloor\cdot\rfloor\cdot$, and $\lceil\cdot\rceil\cdot$, respectively.

II. Principles of B-Spline Interpolation

This section presents the B-spline interpolation theory. First, the traditionally used real-valued scheme is presented, and second, our extension to the complex domain is introduced. This will enable the use of the scheme in the context of radio communications, where complex I/Q signals are utilized.

A. Real-Valued B-Spline Interpolation

Spline interpolation builds on piece-wise polynomials to interpolate between an arbitrary set of points known as control points, under certain continuity and smoothness constraints at the connecting points. With such a piece-wise modeling approach, simpler and lower-order functions can be adopted per individual region, in contrast to the classical polynomial methods where a single high-order expression is utilized to model the whole input range. The use of lower-order functions essentially translates to reduced associated processing complexities [11], [29], as demonstrated in later sections. Also, it allows to better condition the estimation problem [31], i.e., obtaining a lower condition number for the regression matrix and thus avoiding the need of prewhitening/orthogonalization. Another potential advantage of piece-wise models is the ability to use different polynomial orders for different regions, depending on the desired modeling accuracy in each.

To build the piece-wise scheme, we formally define a set of *knots* that divides the input data range into N regions, defined as $\mathbf{T} = \{t_0, t_1, \ldots, t_N\}$, corresponding to regions $i = \{1, 2, \ldots, N\}$. The *knots* are constrained to be non-decreasing, i.e., $t_0 < t_1 < \cdots < t_N$. We consider uniform splines in this paper, since it allows for a simple input-output relation, suitable for adaptive estimation of the control points. Thus, the region width is defined as $\Delta = t_{i+1} - t_i,\ \Delta > 0$.

In general, this construction can be generalized to any number of regions, N, and any spline degree, P. The B-spline segment describing each individual region, $[t_i, t_{i+1})$, is therefore an affine combination of $P+1$ spline curves [32]. Each of these curves is characterized by the Pth-degree spline basis function, given by the De Boor recursion [31] as

$$N_i^P(u) = \frac{u - t_i}{t_{i+P} - t_i} N_i^{P-1}(u) + \frac{t_{i+P+1} - u}{t_{i+P+1} - t_{i+1}} N_{i+1}^{P-1}(u), \tag{2}$$

where the initial 0th-order basis function, $N_i^0(u)$, can be defined as

$$N_i^0(u) = \begin{cases} 1 \text{ if } t_i \le u < t_{i+1}, \\ 0 \text{ otherwise.} \end{cases} \tag{3}$$

The nonzero portions of these segments span over the interval $[t_i, t_{i+P}]$. Additionally, the blending functions N_i^P for different regions are shifted versions of each other, and in general, they can be written as [33]

$$N_i^P(u) = N_0^P(u - i). \tag{4}$$

Consequently, the resulting spline segment in a particular region can be defined as [34]

$$\gamma(u, i) = \sum_{n=i-P-1}^{i-1} N_n^P(u) q_n, \tag{5}$$

where q_n is the corresponding control point multiplying each spline curve. Alternatively, equation in (5) can be expressed as the inner product of two vectors, as

$$\gamma(u,i) = \begin{bmatrix} N_{i-P-1}^P(u) & N_{i-P}^P(u) & \cdots & N_{i-1}^P(u) \end{bmatrix} \begin{bmatrix} q_{i-P-1} \\ q_{i-P} \\ \vdots \\ q_{i-1} \end{bmatrix}. \tag{6}$$

Further developing this expression by substituting the recursion in (2) [28], and combining the spline segments of all the regions, a generic formulation as a function of the spline basis functions is obtained as

$$\gamma(u, i) = \mathbf{\Psi}^T \mathbf{q}, \tag{7}$$

where $\mathbf{q} \in \mathbb{R}^{Q \times 1} = [q_0 \quad q_1 \quad \cdots \quad q_Q]^T$ is the complete set of spline control points. We note that the total number of control points with N regions and spline order P is $Q = N + P$. Additionally, $\mathbf{\Psi}$ reads

$$\mathbf{\Psi} \in \mathbb{R}^{Q \times 1} = [0 \ \cdots \ 0 \ \mathbf{u}^T \mathbf{C}_P \ 0 \ \cdots \ 0]^T. \tag{8}$$

Here, $\mathbf{u}$ is referred to as the abscissa vector, defined as

$$\mathbf{u} \in \mathbb{R}^{(P+1) \times 1} = [u^P \quad u^{P-1} \quad \cdots \quad 1]^T, \tag{9}$$

and $\mathbf{C}_P \in \mathbb{R}^{(P+1) \times (P+1)}$, shown at the bottom of this page, in (1) for $P = 1, \cdots, 4$, is the spline basis matrix, which depends on the chosen spline order P and the knot spacing Δ. The term $\mathbf{u}^T \mathbf{C}_P$ in $\mathbf{\Psi}$ is indexed such that the starting index is i (i.e. $i-1$ leading zeros and $Q - (i+P)$ trailing zeros in (8)), such that only the corresponding control points are contributing to the basis function weighting, according to the region index.

B. Complex-Valued B-Spline Interpolation

In the context of radio communications, complex-valued I/Q signals are utilized. Therefore, real-valued spline interpolation theory needs to be extended to the complex domain. We begin by noting that a memoryless baseband model of a bandpass nonlinear device (such as a PA), with input signal $x[n]$, can be expressed as [35]

$$y[n] = x[n] G(|x[n]|), \tag{10}$$

where $G(|x[n]|) = G_{\mathrm{I}}(|x[n]|) + jG_{\mathrm{Q}}(|x[n]|)$ is the nonlinear complex gain of the device. Notice that $G(|x[n]|)$ depends only on the magnitude of the input signal, and not on its phase.

Let us now denote $x[n]$ as the input signal and $G(|x[n]|)$ as the output of the spline nonlinearity. According to (10), two separate splines can be used to model the I and Q responses. As the input to the splines is a unipolar magnitude signal, the region index i_n and abscissa value u_n, at time instant n, are defined as

$$i_n = \left\lfloor \frac{|x[n]|}{\Delta_{\mathrm{x}}} \right\rfloor + 1, \tag{11}$$

$$u_n = \frac{|x[n]|}{\Delta_{\mathrm{x}}} - (i_n - 1), \tag{12}$$

$$\mathbf{C}_1 = \begin{pmatrix} \frac{-1}{\Delta} & \frac{1}{\Delta} \\ 1 & 0 \end{pmatrix}, \ \mathbf{C}_2 = \frac{1}{2} \begin{pmatrix} \frac{1}{\Delta^2} & \frac{-1}{\Delta^2} & \frac{1}{\Delta^2} \\ \frac{-2}{\Delta} & \frac{2}{\Delta} & 0 \\ 1 & 1 & 0 \end{pmatrix}, \ \mathbf{C}_3 = \frac{1}{6} \begin{pmatrix} \frac{-1}{\Delta^3} & \frac{3}{\Delta^3} & \frac{-3}{\Delta^3} & \frac{1}{\Delta^3} \\ \frac{3}{\Delta^2} & \frac{-6}{\Delta^2} & \frac{3}{\Delta^2} & 0 \\ \frac{-3}{\Delta} & 0 & \frac{3}{\Delta} & 0 \\ 1 & 4 & 1 & 0 \end{pmatrix}, \ \mathbf{C}_4 = \frac{1}{24} \begin{pmatrix} \frac{1}{\Delta^4} & \frac{-4}{\Delta^4} & \frac{6}{\Delta^4} & \frac{-4}{\Delta^4} & \frac{1}{\Delta^4} \\ \frac{-4}{\Delta^3} & \frac{12}{\Delta^3} & \frac{-12}{\Delta^3} & \frac{4}{\Delta^3} & 0 \\ \frac{6}{\Delta^2} & \frac{-6}{\Delta^2} & \frac{-6}{\Delta^2} & \frac{6}{\Delta^2} & 0 \\ \frac{-4}{\Delta} & \frac{-12}{\Delta} & \frac{4}{\Delta} & \frac{12}{\Delta} & 0 \\ 1 & 11 & 11 & 1 & 0 \end{pmatrix}. \tag{1}$$

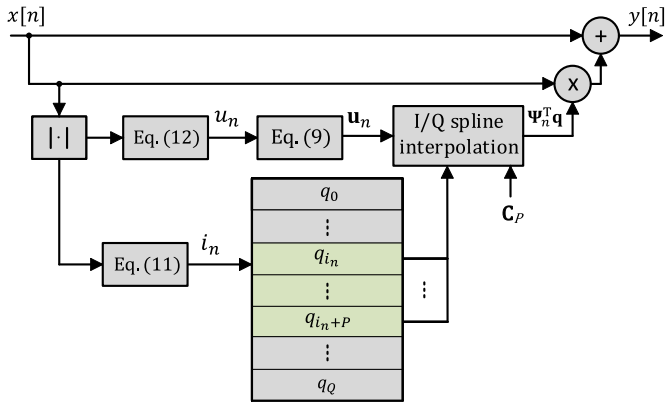


Fig. 1. The injection-based complex spline interpolated LUT scheme presented in (16), utilized in the proposed block-oriented models.

where Δ_{x} is the region width, and u_n is within the interval $[0, \Delta_{\mathrm{x}})$.

The outputs of the two splines modeling the I and Q branch responses of the system can then be written as

$$G_{\mathrm{I}}(|x[n]|) = \mathbf{\Psi}_n^T \mathbf{q}^{\mathrm{re}}, \tag{13}$$

$$G_{\mathrm{Q}}(|x[n]|) = \mathbf{\Psi}_n^T \mathbf{q}^{\mathrm{im}}, \tag{14}$$

where $\mathbf{q}^{\mathrm{re}}$ and $\mathbf{q}^{\mathrm{im}}$ contain the control points for the I and Q splines, and $\mathbf{\Psi}_n$ is defined in (8). Following (10), and using (13) and (14), the spline model output can be written as

$$\begin{aligned} y[n] &= x[n](G_{\mathrm{I}}(|x[n]|) + jG_{\mathrm{Q}}(|x[n]|)) \\ &= x[n]\mathbf{\Psi}_n^T(\mathbf{q}^{\mathrm{re}} + j\mathbf{q}^{\mathrm{im}}) \\ &= x[n]\mathbf{\Psi}_n^T\mathbf{q}, \end{aligned} \tag{15}$$

where $\mathbf{q} = \mathbf{q}^{\mathrm{re}} + j\mathbf{q}^{\mathrm{im}}$ now contains the complex-valued control points.

In this paper, we define the complex nonlinear gain function as a deviation from unity gain, and thus write $G(|x[n]|) = 1 + G_{\mathrm{I}}(|x[n]|) + jG_{\mathrm{Q}}(|x[n]|)$. Thus, with this assumption, the output of the nonlinear subsystem is written as

$$\begin{aligned} y[n] &= x[n] + x[n]\mathbf{\Psi}_n^T\mathbf{q} \\ &= x[n]\mathbf{\Psi}_n^T(\mathbf{1} + \mathbf{q}), \end{aligned} \tag{16}$$

and further illustrated in Fig. 1. Here, $\mathbf{1} \in \mathbb{R}^{Q\times 1}$ denotes a vector of all ones, and the partition of unity property of B-splines [33], or $\mathbf{\Psi}_n^T\mathbf{1} = 1$, is used to arrive to the last form. Applying such a nonlinear subsystem in cascaded models will effectively remove the gain ambiguity between the linear and nonlinear blocks. It will also reduce the dynamic range of the control point vector $\mathbf{q}$, thus requiring less bits in a fixed-point implementation. This will be illustrated in the following subsection.

C. Dynamic Range of $\mathbf{q}$

To illustrate the reduced dynamic range, the Modified Saleh (MS) memoryless PA model presented in [36] is approximated with the spline models drawn in (15) and (16). The MS AM-AM and AM-PM responses can be described by

$$z(r) = \frac{\alpha_z r}{\sqrt{1+\beta_z r^3}}; \quad \phi(r) = \frac{\alpha_\phi}{\sqrt[3]{1+\beta_\phi r^4}} - \epsilon; \tag{17}$$

where r and z represent the magnitudes of the input and output signal, respectively, and ϕ is the phase error of the output signal. Additionally, α_z, β_z, α_ϕ, β_ϕ and ϵ are the AM-AM and AM-PM model coefficients. Herein, it is considered $\alpha_z = 0.82$, $\beta_z = 0.29$, $\alpha_\phi = -0.35$, $\beta_\phi = 1$ and $\epsilon = -0.36$ [36], which were extracted from PA input/output data measured from an LDMOS PA.

An arbitrary OFDM input signal is then passed through the MS model, and a LS algorithm is used to fit both spline techniques to the output model data, thus obtaining the control point vectors from (15) and (16). The magnitude and phase of these values are then quantized with the same fixed number of bits to illustrate the dynamic range reduction. Fig. 2(a) shows the AM-AM response of the MS and the fitted spline models, with both magnitude and phase quantized with 12 bits. It is clearly seen that the modelling accuracy of (15) is reduced when considering the same number of bits as in (16). Additionally, Fig. 2(b) shows the NMSE values of both solutions as the control point vectors are quantized with an increasing number of bits. The modelling performance of (16) is consistently about 18 dB better than that of (15), until the curves essentially saturate. The exact difference depends on the shape of the nonlinear response. However, assessing this further is out of the scope of this paper.

III. Complex Spline-Based Adaptive Hammerstein Model

The adaptive Hammerstein solution aims at identifying an unknown nonlinear system consisting of a memoryless nonlinear function followed by a linear FIR filter [29]. In our previous publication [11], we studied this structure and applied the spline interpolated LUT as the nonlinear block. This model, herein referred to as SPH, serves as a starting point for this paper, and thus we recapitulate the main expressions.

Let us denote by $x[n]$ and $y[n]$ the input and output signals of the SPH model, and by $s[n]$ the intermediate signal after the nonlinear function. Following the B-spline interpolation procedure in (16) for the nonlinear block, and a classical convolution for the FIR filter, the intermediate and output signals of the SPH model can be written as

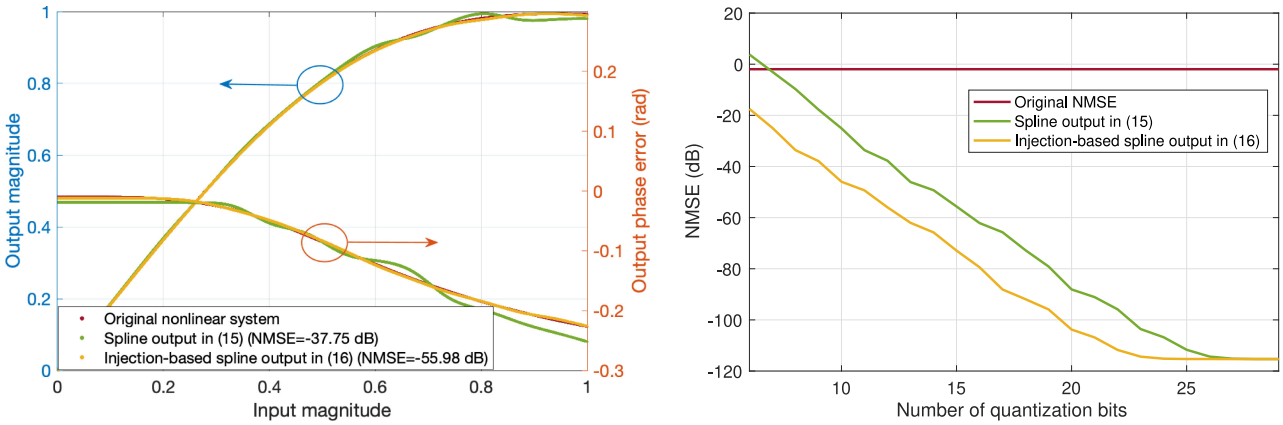
$$\begin{aligned} s[n] &= x[n] + x[n]\mathbf{\Psi}_n^T\mathbf{q}_n, \\ y[n] &= \mathbf{g}_n^T\mathbf{s}_n, \end{aligned} \tag{18}$$

where $\mathbf{g}_n \in \mathbb{C}^{M_{\mathrm{g}}\times 1} = [g[0] \quad g[1] \ \cdots \ g[M_{\mathrm{g}}-1]]^T$, $\mathbf{s}_n \in \mathbb{C}^{M_{\mathrm{g}}\times 1} = [s[n]\, s[n-1] \ \cdots \ s[n-M_{\mathrm{g}}+1]]^T$, and M_{g} denotes the number of taps of the linear filter.

A. Learning Rules and Complexity Analysis

The learning rules for both $\mathbf{g}_n$ and $\mathbf{q}_n$ can be obtained by following the gradient-descent approach. From [11], these expressions read

$$\mathbf{g}_{n+1} = \mathbf{g}_n + \mu_{\mathrm{g}} e[n]\mathbf{s}_n^*, \tag{19}$$



(a) (b)

Fig. 2. Response of the Modified Saleh memoryless nonlinear model and its modelling with (15) and (16) and with 12 quantization bits in (a), and NMSE as a function of the LUT quantization bits in (b).

TABLE I
NUMBER OF REQUIRED ARITHMETIC OPERATIONS IN EACH ITERATION OF THE SPH IDENTIFICATION ALGORITHM, IN TERMS OF FLOPS PER SAMPLE, AND REAL MULTIPLICATIONS PER SAMPLE

Operation		FLOPs/sample	Real multiplications/sample
Model identification	$s[n]$	$2P^2+6P+18$	$P^2+4P+8+sqrt$
	$y[n]$	$8M_\mathrm{g}-2$	$4M_\mathrm{g}$
	Total	$2P^2+6P+8M_\mathrm{g}+16$	$P^2+4P+4M_\mathrm{g}+8+sqrt$
Coefficient updates	$\mathbf{g}_{n+1}$	$8M_\mathrm{g}+2$	$4M_\mathrm{g}+2$
	$\mathbf{q}_{n+1}$	$4P\tau+4P+10\tau+3Q+6$	$4P+2P\tau+6\tau+6$
	Total	$4P\tau+4P+10\tau+8M_\mathrm{g}+3Q+8$	$4P+2P\tau+6\tau+4M_\mathrm{g}+8$
Total		$2P^2+10P+4P\tau+16M_\mathrm{g}+10\tau+3Q+24$	$P^2+8P+2P\tau+6\tau+8M_\mathrm{g}+16+sqrt$

$$\mathbf{q}_{n+1}=\mathbf{q}_n+\mu_\mathrm{q}e[n]\,\boldsymbol{\Xi}_n^T\mathbf{X}_n^*\mathbf{g}_n^*, \tag{20}$$

where $\mathbf{X}_n$ contains the signal regression of $x[n]$ over the span of $\mathbf{g}_n$ in its main diagonal, $\boldsymbol{\Xi}_n\in\mathbb{R}^{M_\mathrm{g}\times Q}=[\boldsymbol{\Psi}_n\ \boldsymbol{\Psi}_{n-1}\ \cdots\ \boldsymbol{\Psi}_{n-M_\mathrm{g}+1}]^T$, and $e[n]=d[n]-y[n]$, $d[n]$ being the observed signal. Additionally, in the learning rule for $\mathbf{q}_{n+1}$, it is assumed that the rate of change of $\mathbf{q}_n$ over the span of the filter length M_g is negligible, i.e., $\mathbf{q}_n\approx\mathbf{q}_{n+M_\mathrm{g}}$. This is a plausible assumption since μ_q is small.

It is important to notice the relatively high complexity involved in calculating the term $\boldsymbol{\Xi}_n^T\mathbf{X}_n^*\mathbf{g}_n^*$ in the update of $\mathbf{q}_{n+1}$, as $\boldsymbol{\Xi}_n^T$ is a $Q\times M_\mathrm{g}$ matrix, somewhat large if $\mathbf{g}_n$ contains a large number of taps. To ease this update, in [11] we proposed a complexity reduction approximation where only a specific temporal span of the matrix $\boldsymbol{\Xi}_n$ (row dimension) is chosen for the update. Obviously, the largest filter taps in $\mathbf{g}_n$ are the ones that contribute the most to the update, and those taps typically correspond to the nearest past samples. Thus, only τ taps, i,e,. rows of $\boldsymbol{\Xi}_n$, $\mathbf{g}_n$, and $\mathbf{x}_n$, can be selected for the learning rule, simplifying the overall update and having little effect in the final modeling performance. This is

$$\boldsymbol{\Xi}_n\in\mathbb{R}^{\tau\times Q}=[\boldsymbol{\Psi}_n\ \ \boldsymbol{\Psi}_{n-1}\ \ \cdots\ \ \boldsymbol{\Psi}_{n-\tau+1}]^T. \tag{21}$$

We refer the reader to [11] for a more detailed presentation of the SPH model.

Finally, we analyze the computational complexity of the SPH model in terms of FLOPs and real multiplications per sample. The results are gathered in Table I. For this complexity analysis, it is assumed that one complex multiplication is calculated with 6 FLOPs or 4 real multiplications, while one complex-real multiplication costs 2 FLOPs or 2 real multiplications, and one real addition costs 1 FLOP and 0 real multiplications. Additionally, one square root costs 8 FLOPs, and it is left indicated as $sqrt$ in the case of real multiplications [16]. Equivalently, one division is left indicated as div in the complexity tables.

We also note that in certain applications, it may not be necessary to update all coefficients at every iteration. One particular example is in the full-duplex scenario, shown in Section VI-A, where the control points $\mathbf{q}_n$ can remain static once steady-state is reached, as the operating conditions of the PA usually do not change drastically once estimated. The filter coefficients, on the other hand, require constant updating, because the self-interference (SI) channel is generally time-varying.

B. Suboptimal Formulation in Scarpiniti Et Al.

As mentioned in the introduction to this paper, the original real-valued presentation of the techniques contained a defect in

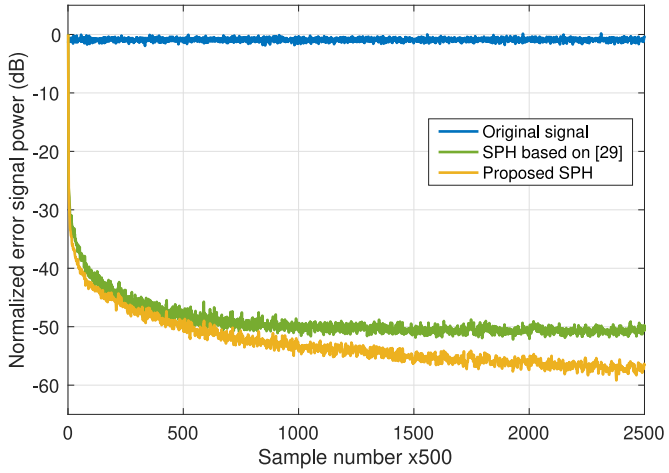


Fig. 3. Convergence comparison between the SPH algorithm based on [29] and the proposed SPH algorithm.

the coefficient updates, which affected the convergence. In [29], the update of the real-valued control point vector $\mathbf{q}_n$ was defined as

$$\mathbf{q}_{i,n+1} = \mathbf{q}_{i,n} + \mu_{\mathrm{q}} e[n]\, \mathbf{C}_P^T \mathbf{U}_{i,n} \mathbf{g}_n. \tag{22}$$

Here, $\mathbf{q}_{i,n} = [q_{i,n}\ q_{i+1,n}\ \cdots\ q_{i+P,n}]^T$ with $q_{k,n}$ denoting the k-th entry of $\mathbf{q}_n$, is the subset of $\mathbf{q}_n$ starting from index i. The matrix $\mathbf{U}_{i,n} = [\mathbf{u}_{i,n}\mathbf{u}_{i,n-1}\ldots\mathbf{u}_{i,n-M+1}]$ collects M past vectors $\mathbf{u}_{i,n-k}$, where each vector assumes the value of $\mathbf{u}_{n-k}$ if evaluated in the same span as the current input sample or in an overlapped span, while otherwise it is a zero vector. The equation in (22) thus only updates the subset $\mathbf{q}_{i,n}$ at time instant n, whereas our correct learning rule in (20) updates the whole vector $\mathbf{q}_n$ in each iteration. The update rule in (22) can also be seen to use some samples from wrong spans to update the current span $\mathbf{q}_{i,n}$, therefore making it suboptimal. To substantiate this, we derived the complex-valued version of (22), which reads

$$\mathbf{q}_{i,n+1} = \mathbf{q}_{i,n} + \mu_{\mathrm{q}} e[n]\, \mathbf{C}_P^T \mathbf{U}_{i,n} \mathbf{X}_n^* \mathbf{g}_n^*, \tag{23}$$

and compared its learning behavior with the update in (20), when learning a cascaded Hammerstein system. Specifically, the Hammerstein structure contains the Saleh nonlinear function presented in Section II-C, followed by an FIR filter with 20 complex-valued taps to account for the memory effects. The same parametrization ($P=3$, $M=20$, $Q=7$, $\mu_{\mathrm{w}}=0.002$, $\mu_{\mathrm{q}}=0.002$) is naturally chosen in both models to ensure fairness, and the resulting error signal powers are then presented in Fig. 3. The proposed Hammerstein solution is seen to clearly outperform the one based on [29] in terms of convergence speed and final steady-state behavior.

We finally note that, although the whole control point vector in (20) is updated in every iteration, only the control points corresponding to the current span (i.e. $q_{i_n}, q_{i_n+1}, \ldots, q_{i_n+P}$) are selected to build the output signal, $y[n]$, in (16). This stems from the definition of $\mathbf{\Psi}_n$, which indexes the term $\mathbf{u}^T\mathbf{C}_P$ starting from index i_n, such that only the correct control points are selected for the interpolation.

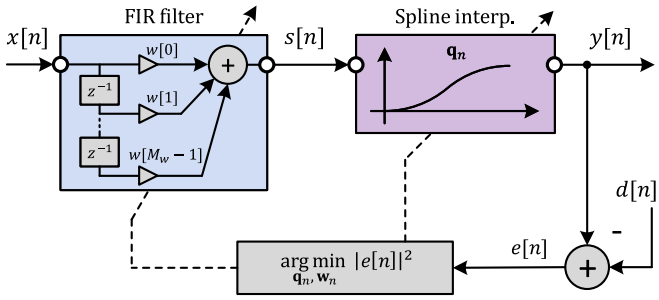


Fig. 4. Architecture of the complex spline interpolated Wiener system. Note that the *spline interpolation* block incorporates the scheme presented in Fig. 1.

IV. Complex Spline-Based Adaptive Wiener Model

In this section, we present the details of the adaptive Wiener solution, herein called SPW, which builds on spline interpolated LUTs. This technique, outlined in Fig. 4, models an unknown nonlinear system with memory effects by cascading a linear filter and a nonlinear function. We denote the linear filter by $\mathbf{w}$, and its tap number by $M_{\mathrm{w}} = M_{\mathrm{w,pre}} + M_{\mathrm{w,post}} + 1$, considering both pre-cursor and post-cursor taps. For simplicity, the model equations consider $M_{\mathrm{w,pre}} = 0$. This consideration is also taken in other following methods.

First, denoting $x[n]$ as the input signal of the model, the intermediate signal $s[n]$ after the linear filter can be written as

$$s[n] = \mathbf{w}_n^T \mathbf{x}_n, \tag{24}$$

where $\mathbf{w}_n \in \mathbb{C}^{M_{\mathrm{w}}\times 1} = [w[0]\ \ w[1]\ \cdots\ w[M_{\mathrm{w}}-1]]^T$, and $\mathbf{x}_n \in \mathbb{C}^{M_{\mathrm{w}}\times 1} = [x[n]\ x[n-1]\ \cdots\ x[n-M_{\mathrm{w}}+1]]^T$.

Secondly, the nonlinear subsystem in (16) is excited with the filter output signal $s[n]$, yielding the SPW model output

$$y[n] = s[n] + s[n]\mathbf{\Psi}_n^T \mathbf{q}_n. \tag{25}$$

A. Learning Rules

In order to make the Wiener model adaptive, two different learning rules, estimating the filter coefficients $\mathbf{w}_n$ and control points $\mathbf{q}_n$, are derived. To this end, we define the error signal

$$e[n] = d[n] - y[n], \tag{26}$$

where the model output signal $y[n]$ is subtracted from the observed signal, or the desired response of the system, denoted here by $d[n]$. The problem lies now in estimating the values of $\mathbf{w}_n$ and $\mathbf{q}_n$ to minimize $e[n]$. This can be done by following the classical gradient-descent solution, where the quantities are learned by following the negative direction of the gradient, towards a local minimum in the mean surface error [37]. The coefficients can be updated iteratively when necessary, depending on the requirements of the final application hosting the algorithm [38].

A cost function can be defined as the instantaneous squared error, depending on both parameters to estimate. In the complex case, the cost function to minimize is

$$J(\mathbf{w}_n, \mathbf{q}_n) = e[n]e^*[n]. \tag{27}$$

Firstly, the general form of the learning rule for the control points reads

$$\mathbf{q}_{n+1} = \mathbf{q}_n - \mu_{\mathrm{q}} \nabla_{\mathbf{q}_n} J(\mathbf{w}_n, \mathbf{q}_n), \tag{28}$$

where $\mu_{\mathrm{q}}[n]$ is the weight or learning rate of the update. Invoking elementary complex differentiation rules [39], while considering the other parameter $\mathbf{w}_n$ constant, the partial derivative can be written as

$$\begin{aligned}\frac{\partial J(\mathbf{w}_n, \mathbf{q}_n)}{\partial \mathbf{q}_n^*} &= e[n]\frac{\partial e^*[n]}{\partial \mathbf{q}_n^*} + e^*[n]\frac{\partial e[n]}{\partial \mathbf{q}_n^*} \\ &= -e[n]\frac{\partial s^*[n]\boldsymbol{\Psi}_n^T(\mathbf{1}+\mathbf{q}_n^*)}{\partial \mathbf{q}_n^*} + 0 \\ &= -e[n]s^*[n]\boldsymbol{\Psi}_n. \end{aligned} \tag{29}$$

Substituting in (28), the final learning rule reads

$$\mathbf{q}_{n+1} = \mathbf{q}_n + \mu_{\mathrm{q}} e[n] s^*[n] \boldsymbol{\Psi}_n. \tag{30}$$

Secondly, the learning rule for the filter $\mathbf{w}_n$ is

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \mu_{\mathrm{w}} \nabla_{\mathbf{w}_n} J(\mathbf{w}_n, \mathbf{q}_n), \tag{31}$$

where $\mu_{\mathrm{w}}[n]$ is the learning rate of the update. The partial derivative with respect to $\mathbf{w}_n$, holding $\mathbf{q}_n$ constant, is given as

$$\begin{aligned}\frac{\partial J(\mathbf{w}_n, \mathbf{q}_n)}{\partial \mathbf{w}_n^*} &= e[n]\frac{\partial e^*[n]}{\partial \mathbf{w}_n^*} + e^*[n]\frac{\partial e[n]}{\partial \mathbf{w}_n^*} \\ &= -e[n]\frac{\partial s^*[n]\boldsymbol{\Psi}_n^T(\mathbf{1}+\mathbf{q}_n^*)}{\partial \mathbf{w}_n^*} + 0 \\ &= -e[n]\boldsymbol{\Psi}_n^T(\mathbf{1}+\mathbf{q}_n^*)\mathbf{x}_n^* \\ &\quad - \frac{s[n]\mathbf{x}_n^*}{|s[n]|}\mathrm{Re}\left\{e[n]\dot{y}[n]\right\}. \end{aligned} \tag{32}$$

Here, $\dot{y}[n] = s^*[n]\dot{\mathbf{u}}_n^T \mathbf{C}_P(\mathbf{1}+\mathbf{q}_{i,n}^*)$, and the vector $\dot{\mathbf{u}}_n \in \mathbb{R}^{(P+1)\times 1} = [P\mathbf{u}_n^{P-1}\ (P-1)\mathbf{u}_n^{P-2}\ \cdots\ 1\ 0]^T$ represents the derivative of the abscissa vector $\mathbf{u}_n$. Substituting this result in (31), the final learning rule reads

$$\begin{aligned}\mathbf{w}_{n+1} &= \mathbf{w}_n + \mu_{\mathrm{w}}\mathbf{x}_n^* \\ &\quad \times \left(e[n]\boldsymbol{\Psi}_n^T(\mathbf{1}+\mathbf{q}_n^*) + \frac{\mathrm{Re}\left\{e[n]\dot{y}[n]s[n]\right\}}{|s[n]|}\right). \end{aligned} \tag{33}$$

B. Complexity of the Proposed SPW Model

As described above, the algorithm is designed to be executed sample by sample, and thus this analysis is presented in terms of FLOPs per sample. This metric basically collects the numbers of multiplications, additions and subtractions to be executed. Furthermore, the number of real multiplications required by the algorithm is also presented separately, as it can constitute an important metric for hardware implementations, such as FPGAs, where resources are limited.

The computational complexity of this method is detailed based on the model identification and coefficient update stages done in each iteration, following the steps below:

1) Filter the input signal to get $s[n]$
2) Compute index and abscissa parameters i_n, u_n, and $\mathbf{u}_n$
3) Perform spline interpolation to obtain $y[n]$
4) Update coefficient $\mathbf{q}_{n+1}$
5) Update coefficient $\mathbf{w}_{n+1}$

The corresponding complexity expressions are collected in Table II , giving the upper bounds for the final complexity, as in a hardware implementation many trivial operations such as multiplications with zero elements or integer powers of 2 or 1/2 do not reflect any added complexity. The complexity expressions are presented as a function of the model parameters P, M_{w} and Q.

V. Complex Spline-Based Adaptive Wiener-Hammerstein Model

The Wiener-Hammerstein model is a combination of the Wiener and Hammerstein models, capable of modeling more complex nonlinear systems with memory. This approach cascades a second linear filter to the Wiener structure to compensate for memory effects appearing after the nonlinearity [30]. Again, the nonlinear function is built upon B-spline interpolation, and it is injected to the linear intermediate signal $l[n]$, as indicated in Fig. 5. We refer to this approach as the SPWH model.

According to the nomenclature in Fig. 5, the intermediate signals, $l[n]$ and $s[n]$, and the model output signal, $y[n]$, can be defined as in (24) and (16). Hence,

$$l[n] = \mathbf{w}_n^T \mathbf{x}_n, \tag{34}$$

$$s[n] = l[n] + l[n]\boldsymbol{\Psi}_n^T \mathbf{q}_n, \tag{35}$$

$$y[n] = \mathbf{g}_n^T \mathbf{s}_n, \tag{36}$$

where the new filters are defined now as $\mathbf{w}_n \in \mathbb{C}^{M_{\mathrm{w}}\times 1} = [w[0]\quad w[1]\ \cdots\ w[M_{\mathrm{w}}-1]]^T$, $\mathbf{g}_n \in \mathbb{C}^{M_{\mathrm{g}}\times 1} = [g[0]\ g[1]\ \cdots\ g[M_{\mathrm{g}}-1]]^T$, and $\mathbf{x}_n$ and $\mathbf{s}_n$ are the regression signals for $x[n]$ and $s[n]$, respectively.

A. Learning Rules

In order to obtain the learning rules for the SPWH model, the same gradient descent approach can be adopted, where an error signal is used to generate a cost function which is minimized with respect to the coefficients to be estimated. In this case, three different coefficient vectors need to be updated, namely the spline control points $\mathbf{q}_n$, and the two linear filters, $\mathbf{w}_n$, and $\mathbf{g}_n$.

The cost function for the SPWH adaptation can then be written as

$$J(\mathbf{w}_n, \mathbf{q}_n, \mathbf{g}_n) = e[n]e^*[n], \tag{39}$$

and the learning expressions are then

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \mu_{\mathrm{w}} \nabla_{\mathbf{w}_n} J(\mathbf{w}_n, \mathbf{q}_n, \mathbf{g}_n), \tag{40}$$

$$\mathbf{q}_{n+1} = \mathbf{q}_n - \mu_{\mathrm{q}} \nabla_{\mathbf{q}_n} J(\mathbf{w}_n, \mathbf{q}_n, \mathbf{g}_n), \tag{41}$$

$$\mathbf{g}_{n+1} = \mathbf{g}_n - \mu_{\mathrm{g}} \nabla_{\mathbf{g}_n} J(\mathbf{w}_n, \mathbf{q}_n, \mathbf{g}_n), \tag{42}$$

where μ_{w}, μ_{q}, and μ_{g} represent the learning rates for each update. The complex gradient approach is again used to calculate the derivatives, assuming that the other coefficients are

TABLE II
NUMBER OF REQUIRED ARITHMETIC OPERATIONS IN EACH ITERATION OF THE SPW IDENTIFICATION ALGORITHM, IN TERMS OF FLOPS PER SAMPLE, AND REAL MULTIPLICATIONS PER SAMPLE

Operation		FLOPs/sample	Real multiplications/sample
Model identification	$s[n]$	$8M_{\mathrm{w}}-2$	$4M_{\mathrm{w}}$
	$y[n]$	$2P^2+7P+18$	$P^2+4P+8+sqrt$
	Total	$2P^2+7P+8M_{\mathrm{w}}+16$	$P^2+4P+4M_{\mathrm{w}}+8+sqrt$
Coefficient update	$\mathbf{w}_{n+1}$	$2P^2+8P+10M_{\mathrm{w}}+17+div$	$P^2+3P+6M_{\mathrm{w}}+15+div$
	$\mathbf{q}_{n+1}$	$2P+2Q+10$	$2P+8$
	Total	$2P^2+10P+2Q+10M_{\mathrm{w}}+27+div$	$P^2+5P+6M_{\mathrm{w}}+23+div$
Total		$4P^2+17P+2Q+18M_{\mathrm{w}}+43+div$	$2P^2+9P+10M_{\mathrm{w}}+31+div+sqrt$

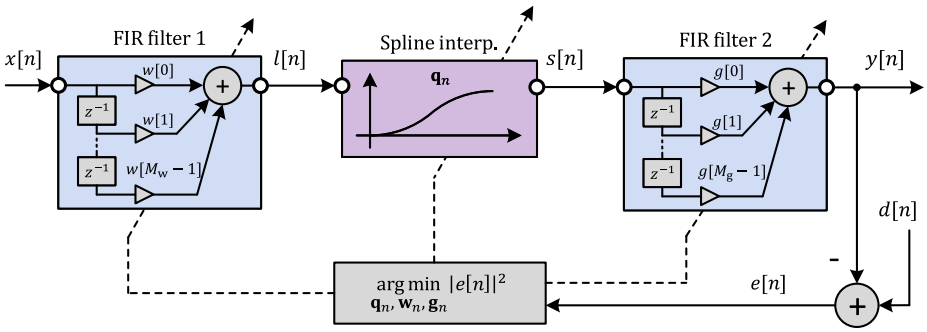


Fig. 5. Architecture of the complex spline interpolated Wiener-Hammerstein system. Note that the *spline interpolation* block incorporates the scheme presented in Fig. 1.

constants. Thus, the first gradient in (40) can be obtained as

$$\begin{aligned}\frac{\partial J(\mathbf{w}_n,\mathbf{q}_n,\mathbf{g}_n)}{\partial \mathbf{w}_n^*} &= e[n]\frac{\partial e^*[n]}{\partial \mathbf{w}_n^*}+e^*[n]\frac{\partial e[n]}{\partial \mathbf{w}_n^*}\\ &= -e[n]\frac{\partial \mathbf{g}_n^H\mathbf{s}_n^*}{\partial \mathbf{w}_n^*}-e^*[n]\frac{\partial \mathbf{g}_n^T\mathbf{s}_n}{\partial \mathbf{w}_n^*}\\ &= \cdots\\ &= e[n]\boldsymbol{\Sigma}_n\mathbf{g}_n^*+\boldsymbol{\Upsilon}_n\mathrm{Re}\left\{e[n]\dot{\mathbf{Y}}_n\mathbf{g}_n^*\right\}, \quad (43)\end{aligned}$$

where $\dot{\mathbf{Y}}_n=\mathrm{diag}\{\dot{y}[n],\ \ldots,\ \dot{y}[n-M_{\mathrm{g}}+1]\}$, and $\boldsymbol{\Sigma}_n$ and $\boldsymbol{\Upsilon}_n$ are presented in (37) and (38), as shown at the bottom of this page, respectively.

The final learning update for $\mathbf{w}_n$ can then be written as

$$\mathbf{w}_{n+1}=\mathbf{w}_n+\mu_{\mathrm{w}}\left(e[n]\boldsymbol{\Sigma}_n\mathbf{g}_n^*+\boldsymbol{\Upsilon}_n\mathrm{Re}\left\{e[n]\dot{\mathbf{Y}}_n\mathbf{g}_n^*\right\}\right), \quad (44)$$

Note that removing the filter $\mathbf{g}_n$ ($\mathbf{g}_n=[1\ 0\ \cdots\ 0]^T$) leads to the learning rule of the SPW model, presented in (44). Additionally, further removing the nonlinearity ($\mathbf{q}_n=[0\ 0\ \cdots\ 0]^T$) leads to the expression of the classical LMS filter, shown in (19).

When considering $\mathbf{g}_n$ and $\mathbf{q}_n$, they directly correspond to the learning rules obtained in the SPH case, due to the SPWH structure. Hence, they can directly be written as

$$\mathbf{g}_{n+1}=\mathbf{g}_n+\mu_{\mathrm{g}}e[n]\mathbf{s}_n^*, \quad (45)$$

$$\mathbf{q}_{n+1}=\mathbf{q}_n+\mu_{\mathrm{q}}e[n]\,\boldsymbol{\Xi}_n^T\mathbf{L}_n^*\mathbf{g}_n^*, \quad (46)$$

where $\mathbf{L}_n$ is a diagonal matrix containing the regression of $l[n]$. Note that in the learning rule for $\mathbf{q}_{n+1}$, it is again assumed that the rate of change of $\mathbf{q}_n$ over the span of the filter length M_{g} is negligible, i.e., $\mathbf{q}_n\approx\mathbf{q}_{n+M_{\mathrm{g}}}$.

B. Complexity of the Proposed SPWH Model

The complexity of the proposed SPWH model is analyzed in terms of FLOPs and real multiplications per sample.

It is again important to remark the relatively high complexity involved in calculating the term $\boldsymbol{\Xi}_n^T\mathbf{L}_n^*\mathbf{w}_n^*$ in the update of $\mathbf{q}_{n+1}$, as $\boldsymbol{\Xi}_n$ is a $Q\times M_{\mathrm{w}}$ matrix, and $\mathbf{w}_n$ may contain a large number of taps. To ease this update, we propose a similar complexity reduction approximation as was done for the SPH model, where only a specific temporal span of the matrix $\boldsymbol{\Xi}_n$ (row dimension) is chosen for the update. Thus, only τ taps, i,e,. rows of $\boldsymbol{\Xi}_n$, $\mathbf{w}_n$, and $\mathbf{L}_n$ can be selected for the learning rule, simplifying the overall update while having minimal effect on the performance. The approximation is similar to the SPH model, and is given in (21).

$$\boldsymbol{\Sigma}_n=\left[\boldsymbol{\Psi}_n(\mathbf{1}+\mathbf{q}_n^*)\mathbf{x}_n^*\quad \boldsymbol{\Psi}_{n-1}(\mathbf{1}+\mathbf{q}_{n-1}^*)\mathbf{x}_{n-1}^*\quad \cdots\quad \boldsymbol{\Psi}_{n-M_{\mathrm{g}}+1}(\mathbf{1}+\mathbf{q}_{n-M_{\mathrm{g}}+1}^*)\mathbf{x}_{n-M_{\mathrm{g}}+1}^*\right], \quad (37)$$

$$\boldsymbol{\Upsilon}_n=\left[\frac{l[n]\,\mathbf{x}_n^*}{|l[n]|}\quad \frac{l[n-1]\,\mathbf{x}_{n-1}^*}{|l[n-1]|}\quad \cdots\quad \frac{l[n-M_{\mathrm{g}}+1]\,\mathbf{x}_{n-M_{\mathrm{g}}+1}^*}{|l[n-M_{\mathrm{g}}+1]|}\right]. \quad (38)$$

TABLE III
NUMBER OF REQUIRED ARITHMETIC OPERATIONS IN EACH ITERATION OF THE SPWH IDENTIFICATION ALGORITHM, IN TERMS OF FLOPS PER SAMPLE, AND REAL MULTIPLICATIONS PER SAMPLE

Operation		FLOPs/sample	Real multiplications/sample
Model identification	$l[n]$	$8M_{\mathrm{w}} - 2$	$4M_{\mathrm{w}}$
	$s[n]$	$2P^2 + 7P + 19$	$P^2 + 4P + 8 + sqrt$
	$y[n]$	$8M_{\mathrm{g}} - 2$	$4M_{\mathrm{g}}$
	Total	$2P^2 + 7P + 8(M_{\mathrm{w}} + M_{\mathrm{g}}) + 15$	$P^2 + 4P + 4(M_{\mathrm{w}} + M_{\mathrm{g}}) + 8 + sqrt$
Coefficient updates	$\mathbf{w}_{n+1}$	$2P^2 + 6P + 12M_{\mathrm{w}}M_{\mathrm{g}} + 20M_{\mathrm{w}} + 14M_{\mathrm{g}} + 7 + div$	$P^2 + 3P + 6M_{\mathrm{w}}M_{\mathrm{g}} + 14M_{\mathrm{w}} + 8M_{\mathrm{g}} + 9 + div$
	$\mathbf{q}_{n+1}$	$4\tau Q + 6\tau + 6Q + 2$	$4P + 2P\tau + 6\tau + 6$
	$\mathbf{g}_{n+1}$	$8M_{\mathrm{g}} + 2$	$4M_{\mathrm{g}} + 2$
	Total	$2P^2 + 6P + 4\tau Q + 6\tau + 6Q$ $+12M_{\mathrm{w}}M_{\mathrm{g}} + 20M_{\mathrm{w}} + 22M_{\mathrm{g}} + 11$	$P^2 + 7P + 2P\tau + 6\tau + 6M_{\mathrm{w}}M_{\mathrm{g}}$ $+14M_{\mathrm{w}} + 12M_{\mathrm{g}} + 17 + div$
Total	-	$4P^2 + 13P + 4\tau Q + 6\tau + 6Q$ $+12M_{\mathrm{w}}M_{\mathrm{g}} + 28M_{\mathrm{w}} + 30M_{\mathrm{g}} + 26 + div$	$2P^2 + 11P + 2P\tau + 6\tau + 6M_{\mathrm{w}}M_{\mathrm{g}}$ $+18M_{\mathrm{w}} + 16M_{\mathrm{g}} + 25 + sqrt + div$

The upper bound complexity expressions for the SPWH model are gathered in Table III. Note that the case $\tau = M_{\mathrm{w}}$ corresponds to the original learning update, with no complexity reduction method applied.

VI. APPLICATIONS, USE CASES, AND RF MEASUREMENTS

The proposed modeling approaches can be applied to many use cases and applications. In this section, we specifically focus on DSIC and DPD as direct and inverse modeling problems, respectively. We present extensive RF measurement results to verify and demonstrate the performance of the proposed techniques.

In the results, the widely used MP and GMP models are adopted for reference, as they are the most common high-performance techniques used in the literature [9], [16], [23], [40]. The input/output relation of the GMP model can be expressed as

$$y[n] = \sum_{\substack{p=1 \\ p\ \mathrm{odd}}}^{P_{\mathrm{MP}}} \sum_{m=0}^{M_{\mathrm{MP}}} \alpha_{p,m} x[n-m]|x[n-m]|^{p-1} + \sum_{\substack{p=3 \\ p\ \mathrm{odd}}}^{P_{\mathrm{C}}} \sum_{m=0}^{M_{\mathrm{C}}} \sum_{\substack{l=-L_{\mathrm{C}} \\ l\neq 0}}^{L_{\mathrm{C}}} \beta_{p,m,l} x[n-m]|x[n-m+l]|^{(p-1)}, \tag{47}$$

where P_{MP}, M_{MP}, and $\alpha_{p,m}$ represent the nonlinear order, memory, and model coefficients of the aligned envelope terms. Additionally, P_{C}, M_{C}, L_{C}, and $\beta_{p,m,l}$ are the nonlinear order, memory, depth, and model coefficients of the lagging and leading terms. For simplicity, we assume in this work the same number of lagging and leading coefficients, denoted here by $C = \lfloor P_{\mathrm{C}}/2 \rfloor M_{\mathrm{C}} L_{\mathrm{C}}$. In fact, C indicates the number of additional coefficients compared to the MP model. Obviously, the MP model can be directly obtained from (47) by setting the lagging and leading coefficients to zero (and thus $C = 0$). In both models, the set of coefficients can be learned using the LMS algorithm, thus facilitating a fair comparison with the proposed algorithms. The basis functions of the models, however, need to be prewhitened or orthogonalized before the processing [41], if LMS is applied. The reason for this is the poor convergence performance of the LMS algorithm if the elements of the input vector are highly correlated, stemming from the large eigenvalue spread of the input signal covariance matrix. In this case, the static basis functions can certainly be expected to be correlated, since they all depend on the original transmit signal. Orthogonalizing them is thus a necessary step to ensure efficient parameter learning by the LMS algorithm. The computational complexity of this process is included within the parameter update stage. Further details and analysis of the orthogonalized polynomial models and LMS learning in the context of DSIC can be found in [25].

A. Direct Modeling: DSIC in IBFD Transceivers

Self-interference cancellation technology has gathered a lot of attention in the past 10 years as a key technology for realizing IBFD communications [3], [23], [40]. IBFD devices are capable of transmitting and receiving simultaneously on the same channel, thus enabling a twofold increase in the spectral efficiency and data rate, without requiring any additional bandwidth. Fig. 6 shows a typical IBFD transceiver structure. Besides IBFD, DSIC technology can benefit also other communication technologies [4], as well as joint sensing and communications [42].

To make IBFD commercially viable, the problem of SI must be dealt with. SI refers to the unwanted own transmit signal that is leaked into the receiver chain, potentially saturating the receiver and masking the desired received signal. Thus, any full-duplex transceiver must be capable of removing the SI signal in an efficient manner to provide reliable communication. In practice, this is done in three stages: using propagation domain isolation techniques, analog SI cancellation, and digital cancellation (DSIC), as illustrated in Fig. 6. DSIC, which is our focus here, is a forward modeling problem, typically aiming at modeling the nonlinear transmitter and the SI propagation channel [5]. The proposed cascaded models are well suited for modeling such a system.

We demonstrate the feasibility of the proposed techniques in the context of DSIC in IBFD by utilizing measured data

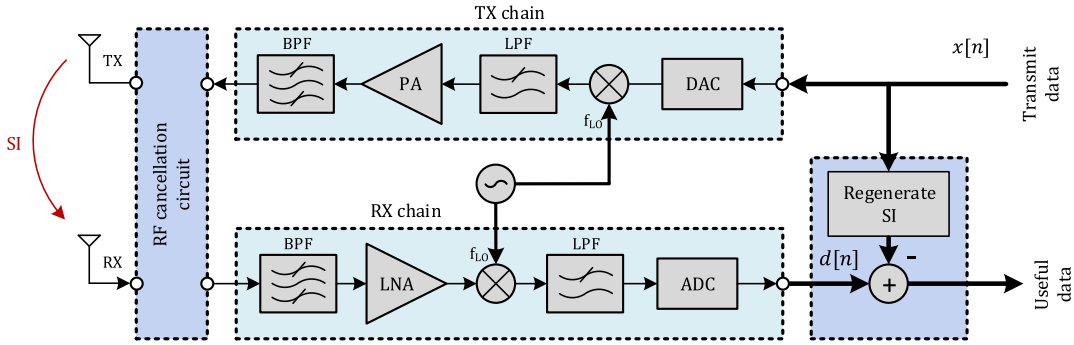


Fig. 6. Generic IBFD transceiver architecture.

TABLE IV
THE RF MEASUREMENT AND MODEL PARAMETERS IN THE DSIC EXPERIMENT. THE COMPUTATIONAL COMPLEXITY IN THE CANCELLATION AND PARAMETER UPDATE STAGES IS PRESENTED IN THE LAST TWO COLUMNS, IN TERMS OF REAL MULTIPLICATIONS PER SAMPLE. THE MP AND GMP BF ORTHOGONALIZATION COMPLEXITY IS INCLUDED IN THE COEFFICIENT UPDATE STAGE

System parameters	**Value**	**Model**	P	C	Q	$M_{\mathrm{w,pre}}/M_{\mathrm{w,post}}$	$M_{\mathrm{g,pre}}/M_{\mathrm{g,post}}$	**Mul. - canc.**	**Mul. - update**
Center frequency	2.56 GHz	SPH DSIC	3	-	5	-	25 / 50	329	380
Transmit waveform	20/40/80 MHz OFDM	SPW DSIC	3	-	5	25 / 50	-	329	497
Meas. sample size	1 Msamples	SPWH DSIC	3	-	5	0 / 10	25 / 50	369	5647
Transmit power	$\approx$ 24 dBm	MP DSIC	11	0	-	-	25 / 50	1840	833 570
RX losses	4 dB	GMP DSIC	11	12	-	-	25 / 50	1960	923 522

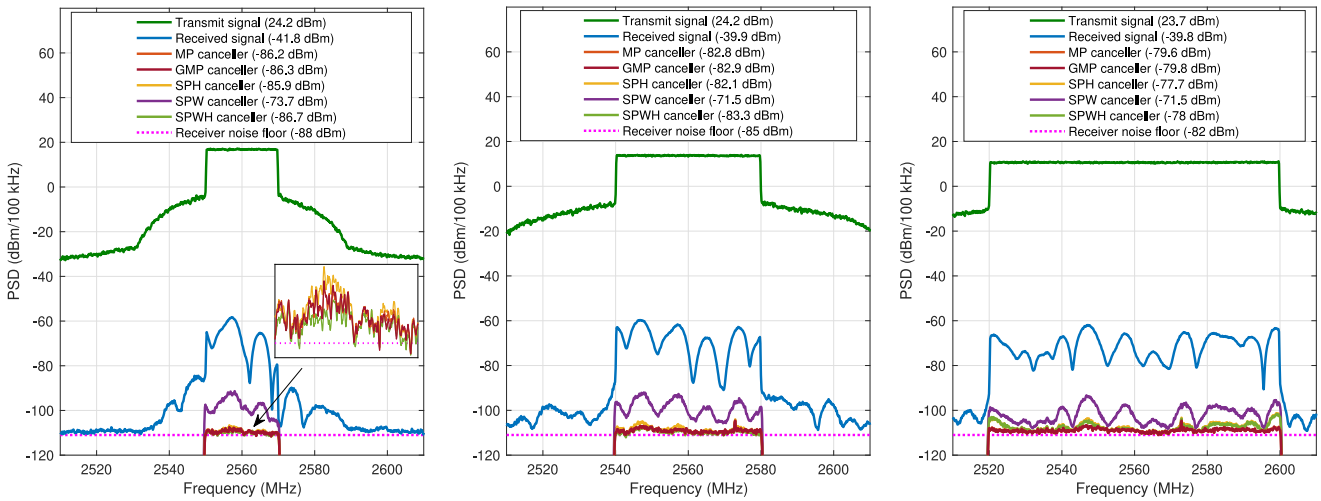


Fig. 7. PSDs of the overall signal after the different DSICs, for (a) 20 MHz, (b) 40 MHz, and (c) 80 MHz instantaneous bandwidths.

originally used in [43], where a back-to-back relay antenna was used to provide the analog isolation between the transmitter and receiver nodes, and no active analog cancellation was used.

The parameterizations for the different DSIC models are collected in Table IV, along with the computational complexity in terms of real multiplications per input sample in the cancellation and coefficient update stages. In both SPH and SPWH models, the proposed complexity reduction method is considered, with $\tau = 5$. In the GMP model, $P_{\mathrm{C}} = 5$, $M_{\mathrm{C}} = 3$, and $L_{\mathrm{C}} = 2$ are considered, which leads to $C = 12$ coefficients. Additionally, a large number of memory taps is considered in each model due to the high frequency selectivity of the SI channel. First, the power spectral densities (PSDs) of the transmit, received, and signal after cancellation with the algorithms are shown in Fig. 7, for three different instantaneous signal bandwidths of 20, 40, and 80 MHz. In all cases, the SPH and SPWH models achieve a similar cancellation to that of the MP and GMP models, regardless of the substantial complexity reduction. With the narrowest 20 MHz bandwidth, the amounts of achieved digital cancellation are 44.1 dB (SPH) and 44.9 dB (SPWH). With the widest 80 MHz bandwidth, the amounts of achieved cancellation are 37.9 dB (SPH) and 38.2 dB (SPWH). These

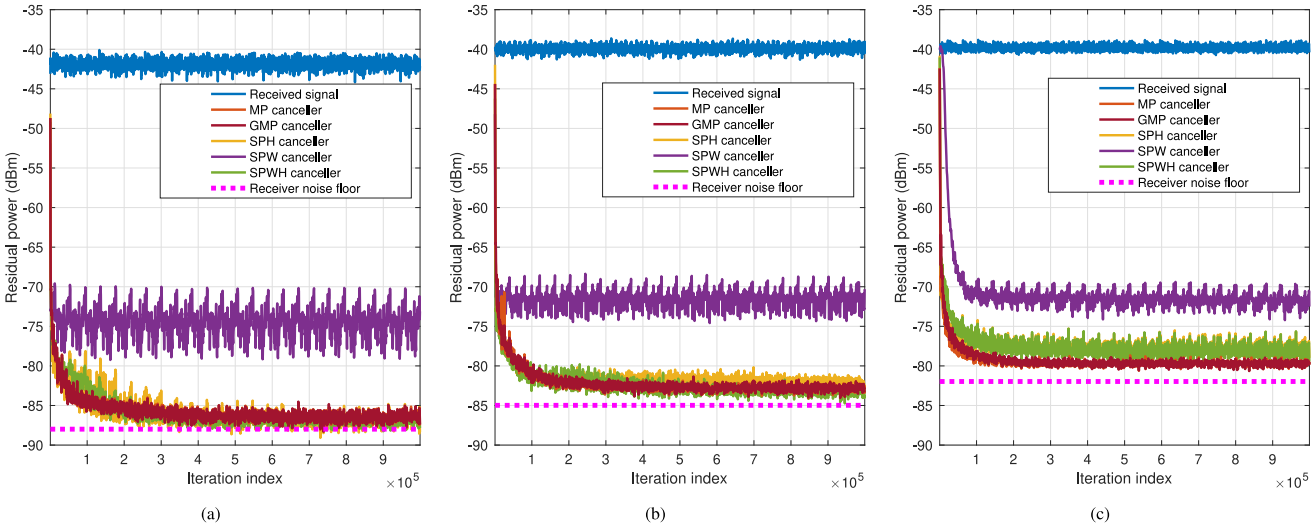


Fig. 8. Residual powers with respect to the iteration index after the different DSICs, for (a) 20 MHz, (b) 40 MHz, and (c) 80 MHz instantaneous bandwidths.

results indicate high modeling accuracy, with the SPWH model even outperforming the MP and GMP models in the narrowband cases. In this particular measurement, the SPW model yields somewhat lower performance with around 32 dB cancellation in the three considered bandwidth cases. This is expected due to the SI signal model naturally following a Hammerstein type structure. Additionally, the proposed methods are capable of reducing the cancellation complexity by more than 82% (SPH and SPW) and 80% (SPWH) in the cancellation stage with respect to the MP and GMP models. In the coefficient update stage, the computational complexity of all the methods is reduced by more than 99% compared to the MP and GMP. This is due, in large part, to them not requiring BF orthogonalization, in contrast to the polynomial-based models. These results illustrate the excellent performance-complexity trade-off of the proposed solutions.

In order to evaluate the convergence properties of the algorithms, Fig. 8 shows the residual powers of the received signals after cancellation, for the three bandwidths mentioned above. All the models are essentially achieving a similar convergence speed, despite the proposed spline based models not using orthogonalization. Altogether, the results show that excellent digital cancellation can be obtained with the proposed algorithms, regardless of the large processing complexity reduction.

B. Inverse Modeling: Digital Predistortion

In this section, the proposed models are tested in the context of DPD. DPD is a well-established technique that aims at minimizing the unwanted emissions and nonlinear distortion originated from PAs. This approach applies a nonlinear transformation to the input signal that pre-compensates for the unwanted PA effects [2]. Thus, in DPD processing, choosing a nonlinear model with the best complexity-performance trade-off for the device at hand is a key design challenge.

In this paper, we apply DPD linearization to a millimeter wave (mmW) active antenna array system operating at 28 GHz. The array linearization problem, in the single user case, is a single-input single-output identification problem, similar to regular DPD linearization of a single PA [44]–[46]. However, in a K-antenna array, there are also K PAs, each with unique (though somewhat similar) nonlinear characteristics. The DPD can thus guarantee good linearization only in the main beam direction, while in the rest of the directions, the beampattern of the array will help to keep the nonlinear distortions low [44]. Another important feature of arrays is the load modulation of the PAs, which occurs due to coupling between the antennas [47]. This will make the nonlinear characteristics of the array *beam-dependent*. Thus, any linearization solution for active arrays needs to take this into account, as the optimal DPD will also depend on the beam direction. DPD solutions which are real-time adaptive are one viable solution to track and adapt the DPD coefficients as the beam is steered. A further concern at mmWs, as the signal bandwidths are wide and matching is more problematic compared to lower frequencies, is the frequency selectivity of the transmitter. There could be, for example, distinct frequency selectivity between the baseband/IF parts and the PA, which the DPD estimator would interpret as part of the PA response, thus requiring large memory depth and unnecessary complexity in the DPD. The proposed adaptive cascaded models can *separate* these linear distortions from the nonlinear effects of the PAs, and thus offer much lower complexity linearization, which is an important criterion when designing real-time systems.

For learning, we adopt the so-called ILA, which is illustrated in Fig. 9. Here, the DPD coefficients are estimated through the post-inverse of the PA, which can be calculated directly from the PA input and output signals [9]. The post-inverse coefficients are copied to the digital predistorter, and the learning procedure is then typically iterated a few times to reach steady-state.

The DPD performance is evaluated through the well-known error vector magnitude (EVM) and normalized mean square error (NMSE) [16]. However, since an over-the-air (OTA) DPD system is considered, the out-of-band performance is measured with the total radiated power (TRP) based adjacent channel

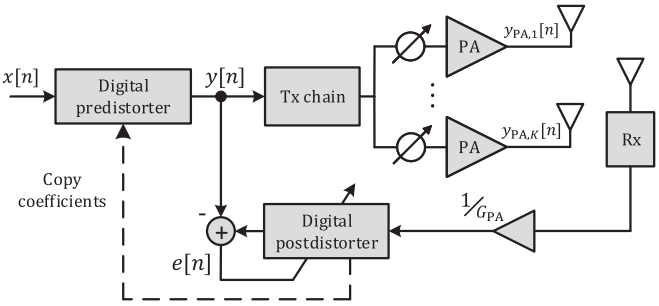


Fig. 9. Illustration of the ILA DPD system, in the context of a K-antenna array. The DPD is estimated directly from the predistorted and OTA received signals. G_{PA} represents the estimate of the complex linear gain of the Tx chain.

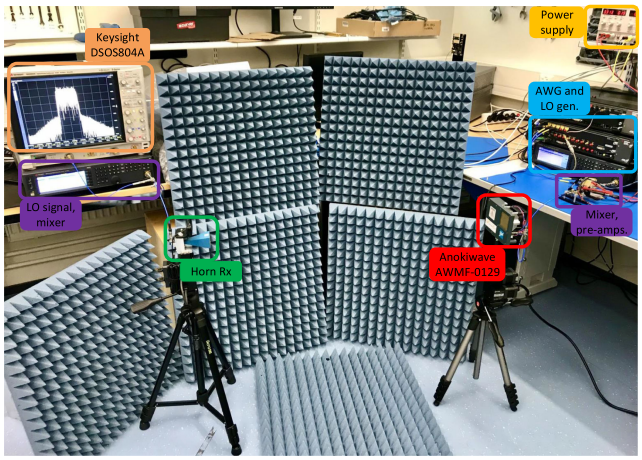


Fig. 10. The 5 G NR mmW/FR-2 RF deployment used in the DPD experiment.

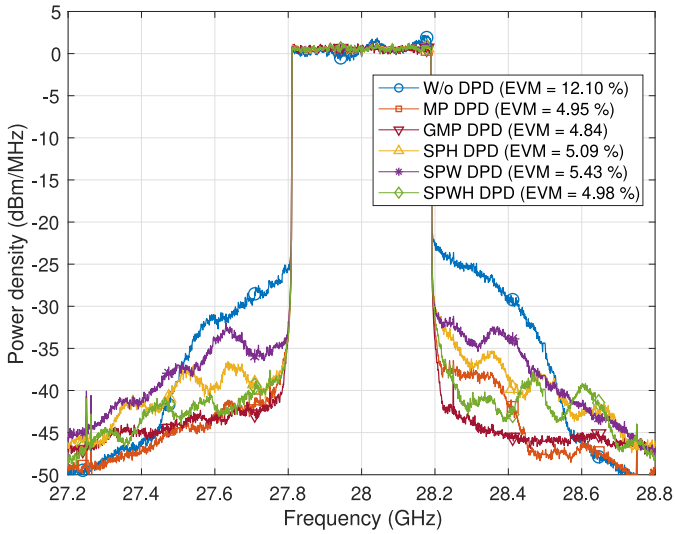


Fig. 11. Example illustration of DPD OTA linearization results at 28 GHz, with 400 MHz bandwidth and EIRP $\approx$ +42 dBm.

leakage ratio (ACLR), which is the filtered mean power centered on the assigned channel frequency to the filtered mean power centred on an adjacent channel frequency, measured by integrating the powers over the whole beamspace, while keeping the *beamforming angle* fixed [48].

The overall array measurement setup is shown in Fig. 10, containing the test device, which is an Anokiwave AWMF-0129 64-element active antenna array, together with other instruments to facilitate signal generation and analysis at 28 GHz center frequency. Firstly, a Keysight M8190 arbitrary waveform generator (AWG) is used to generate an IF signal at 3.5 GHz. The signal is then upconverted to the 28 GHz carrier by a Keysight N5183B-MXG acting as the LO at 24.5 GHz and a Marki Microwave T31040 L mixer, and then filtered by a Marki microwave FB3300 band-pass filter (BPF). The upconverted signal is then driven through two pre-amplifiers, Analog Devices HMC499LC4 and Analog Devices HMC1131, facilitating enough power to drive the active antenna array to saturation. The array transmits the signal OTA, and the signal is captured by a horn antenna at the observation receiver, such that the receiver antenna is well aligned with the main beam. In this experiment, the *beamforming angle* is considered as 0 degrees. The signal is then carried through an attenuator and another mixer to be down-converted back to IF. Then, a Keysight DSOS804 A oscilloscope is used as the actual digitizer to facilitate the post-processing on a host PC, where the DPD algorithms are executed.

In the DPD measurements, a 5G NR FR-2 OFDM signal with 120 kHz subcarrier spacing (SCS) and 264 active resource blocks (RBs) is adopted. This configuration maps to 400 MHz channel bandwidth. The proposed DPD models are tested with 12 ILA iterations, and 50 ksamples each. The adopted model parametrization and DPD complexity are presented in Table V. The cross-terms of the GMP model are configured with $P_{\mathrm{C}} = 9$, $M_{\mathrm{C}} = 2$, $L_{\mathrm{C}} = 2$, which leads to $C = 16$ coefficients. A snap-shot linearization example is showed in Fig. 11, at effective isotropic radiated power (EIRP) of +42 dBm. Within the three proposed solutions, the SPWH model obtains the best linearization performance, and the closest to the reference MP and GMP solutions. This model is followed by the SPH DPD, which also obtains a good DPD linearization, slightly degraded but close to SPWH. The SPW model follows somewhat behind the other DPD solutions, but it is still capable of obtaining a fair amount of performance. This result can be explained as follows. On the one hand, the SPW model cannot correct memory effects and distortion appearing in the mixing and filtering stage (T31040 L MX and FB3300 BPF), before the driver PA, leading to performance loss. On the other hand, the SPH model cannot properly account for the frequency selectivity appearing in the OTA transmission. The SPWH model, in turn, is able to account for both phenomena, achieving thus an enhanced performance compared to the other models.

The same conclusion can be made from Fig. 12, which shows the NMSE and TRP ACLR performance as a function of the iteration number. It can also be observed that the convergence of the DPD models is fast in general, reaching the steady-state in approximately 4-5 DPD iterations. The 5G NR ACLR limit of 28 dBc, measured using the TRP [48] approach, is fulfilled in all cases. It can be seen from Table V that the computational complexity with respect to the classical MP is also greatly

TABLE V
THE RF MEASUREMENT AND MODEL PARAMETERS IN THE DPD EXPERIMENT. THE COMPUTATIONAL COMPLEXITY OF THE DPD MAIN PATH AND DPD LEARNING STAGES IS PRESENTED IN THE LAST TWO COLUMNS, IN TERMS OF REAL MULTIPLICATIONS PER SAMPLE. THE MP AND GMP BF ORTHOGONALIZATION COMPLEXITY IS INCLUDED IN THE DPD LEARNING STAGE

System parameters	Value	SI model	P	C	Q	$M_{\mathrm{w,pre}}/M_{\mathrm{w,post}}$	$M_{\mathrm{g,pre}}/M_{\mathrm{g,post}}$	Mul. - main path	Mul. - learning
Center frequency	28 GHz	SPH DPD	3	-	5	-	0 / 15	89	140
Transmit waveform	400 MHz NR OFDM	SPW DPD	3	-	5	0 / 15	-	89	137
SCS / RBs	120 kHz / 264	SPWH DPD	3	-	5	0 / 13	0 / 13	133	1459
PAPR (@ 0.01%)	7 dB	MP DPD	11	0	-	-	0 / 10	280	17 970
EIRP	+42 dBm	GMP DPD	11	16	-	-	0 / 10	400	33 162

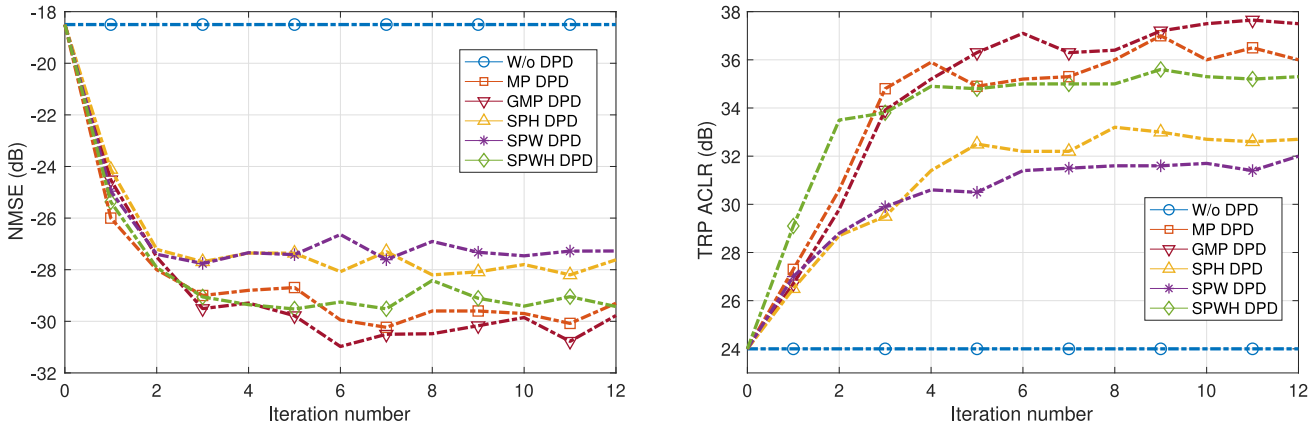


Fig. 12. DPD performance in terms of (a) NMSE, and (b) TRP ACLR of the proposed DPD models as a function of the iteration index, with 400 MHz bandwidth and EIRP ≈ +42 dBm.

reduced by 68% (SPH and SPW) and 53% (SPWH) in the DPD main path. With respect to GMP, the complexity is reduced by 78% (SPH and SPW) and 67% (SPWH). In the DPD learning, the computational complexity of the proposed models is reduced by more than 90% with respect to MP, and by more than 95% with respect to GMP. These results verify the excellent performance-complexity trade-off of the proposed techniques.

To further demonstrate the performance of the proposed models, a third and final experiment featuring a power sweep is presented. The power sweep shows the ACLR, measured through the TRP, and EVM performance metrics as a function of the EIRP, which ranges from +38.5 dBm to +42.5 dBm. With this experiment, we try to shed some light into two main things. First, to evaluate whether the TRP ACLR or the EVM is the limiting performance metric in the mmW system, in terms of maximum achievable EIRP. Second, to assess the performance of the proposed and reference DPD models as the radiated power of the antenna array varies. The obtained TRP ACLR and EVM results are presented in Fig. 13. Firstly, it can be clearly seen that, when no predistortion is applied, the EVM constitutes the limiting factor in terms of maximum achievable EIRP, basically surpassing the 8% EVM limit [48] at approximately +39.8 dBm, where the TRP ACLR limit is still fulfilled. However, when the proposed predistortion techniques are applied, both EVM and TRP ACLR limits are satisfied at least up until +42.5 dBm, and clearly somewhat beyond, as the figure shows. These findings indicate an increase in the overall power efficiency of the transmit chain, as the antenna array can be operated closer to saturation thanks to the DPD operation. Secondly, the power sweep also indicates that the DPD algorithms behave in a similar manner as commented before. The proposed SPWH model is capable of achieving a similar linearization performance to that of the MP and GMP models, despite the reduced processing and learning complexities. It is then followed by the SPH and SPW models, whose performance lie somewhat behind, but still fulfilling the specified 3GPP limits. In general, the proposed algorithms achieve good amounts of linearization performance, and successfully satisfy the 3GPP specifications [48] at least up to EIRP of +42.5 dBm and clearly beyond, while allowing for excellent processing and learning complexity reductions. These results further confirm the very favorable complexity-performance trade-offs of the proposed solutions in different scenarios.

VII. CONCLUSION

In this paper, a class of low-complexity cascaded models and learning algorithms were proposed for modeling of complex-valued nonlinear systems with memory. Hammerstein, Wiener, and Wiener-Hammerstein models were constructed, building on uniform spline-interpolated LUTs as the elementary nonlinear functions and FIR filters for the linear blocks, and gradient descent based adaptive algorithms were derived for each of the models. This approach was shown to yield remarkably simple

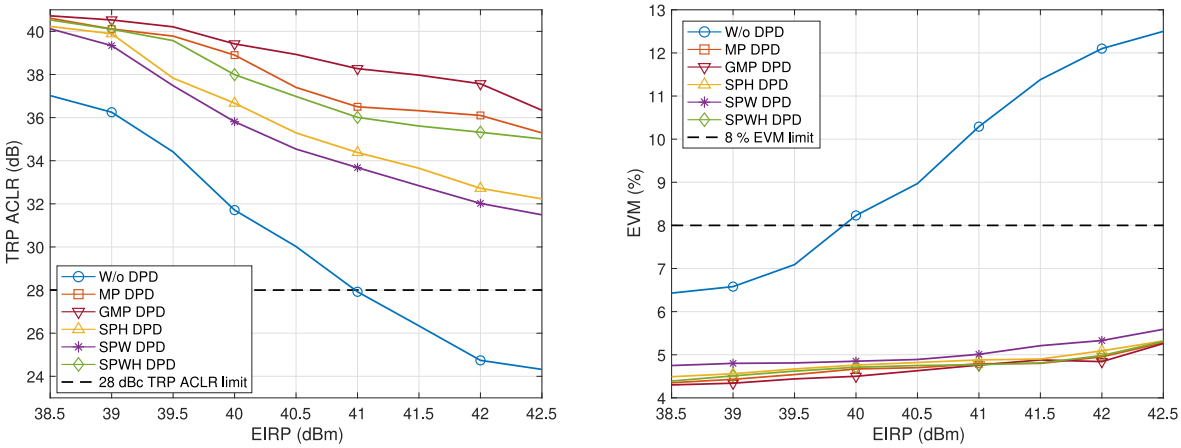


Fig. 13. DPD OTA linearization performance at 28 GHz with 400 MHz bandwidth, as a function of the EIRP of the proposed DPD models in terms of (a) TRP ACLR, and (b) EVM.

yet effective solutions, suitable for real-time implementation and continuous learning. In order to validate and verify the proposed methods, RF measurements were carried out in the context of full-duplex self-interference cancellation and digital predistortion. First, the received signal from a complete full-duplex prototype environment was used to test the SI cancellation capability of the proposed methods, achieving high cancellation levels with much lower complexity compared to state-of-the-art. Second, a millimeter-wave active antenna array operating at 28 GHz was linearized with the proposed techniques, reducing the unwanted emissions to levels below the specified limits. These results illustrate the excellent complexity-performance trade-off of the proposed methods, as they can offer modeling performance very close to state-of-the-art regardless of the substantial complexity reduction.

REFERENCES

[1] S. C. Cripps, *RF Power Amplifiers for Wireless Communications*, 2nd ed. Norwood, MA, USA: Artech House, 2006.

[2] F. M. Ghannouchi and O. Hammi, "Behavioral modeling and predistortion," *IEEE Microw. Mag.*, vol. 10, no. 7, pp. 52–64, Dec. 2009.

[3] A. Sabharwal, P. Schniter, D. Guo, D. W. Bliss, S. Rangarajan, and R. Wichman, "In-band full-duplex wireless: Challenges and opportunities," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 9, pp. 1637–1652, Sep. 2014.

[4] S. Hong *et al.*, "Applications of self-interference cancellation in 5G and beyond," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 114–121, Feb. 2014.

[5] D. Korpi, "Full-duplex wireless: Self-interference modeling, digital cancellation, and system studies," Ph.D. dissertation, Lab. Electron. Commun. Eng., Tampere Univ. Technol., Dec. 2017.

[6] S. Benedetto and E. Biglieri, "Nonlinear equalization of digital satellite channels," *IEEE J. Sel. Areas Commun.*, vol. SAC-1, no. 1, pp. 57–62, Jan. 1983.

[7] M. Ibnkahla, N. Bershad, J. Sombrin, and F. Castanie, "Neural network modeling and identification of nonlinear channels with memory: Algorithms, applications, and analytic models," *IEEE Trans. Signal Process.*, vol. 46, no. 5, pp. 1208–1220, 1998.

[8] A. Zhu, J. C. Pedro, and T. C. 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] 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.

[10] T. Liu, S. Boumaiza, and F. M. Ghannouchi, "Dynamic behavioral modeling of 3G power amplifiers using real-valued time-delay neural networks," *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 3, pp. 1025–1033, Mar. 2004.

[11] P. Pascual Campo, D. Korpi, L. Anttila, and M. Valkama, "Nonlinear digital cancellation in full-duplex devices using spline-based hammerstein model," in *Proc. IEEE Global Commun. Conf.*, Dec. 2018.

[12] L. Ding, R. Raich, and G. Tong Zhou, "A hammerstein predistortion linearization design based on the indirect learning architecture," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, vol. 3, May 2002, pp. 2689–2692.

[13] X. Hong, Yu Gong, and Sheng Chen, "A wiener model for memory high power amplifiers using b-spline function approximation," in *Proc. 17th Int. Conf. Digit. Signal Process.*, Jul. 2011, pp. 1–5.

[14] X. Wu, N. Zheng, X. Yang, J. Shi, and H. Chen, "A spline-based hammerstein predistortion for 3 G power amplifiers with hard nonlinearities," in *Proc. 2nd Int. Conf. Future Comput. Commun.*, vol. 3, May 2010, pp. 741–745.

[15] S. Chen, X. Hong, J. Gao, and C. J. Harris, "Complex-valued b-spline neural networks for modeling and inverting hammerstein systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 9, pp. 1673–1685, Sep. 2014.

[16] 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.

[17] X. Hong and S. Chen, "Modeling of complex-valued wiener systems using b-spline neural network," *IEEE Trans. Neural Netw.*, vol. 22, no. 5, pp. 818–825, May 2011.

[18] T. Liu, S. Boumaiza, and F. M. Ghannouchi, "Augmented hammerstein predistorter for linearization of broad-band wireless transmitters," *IEEE Trans. Microw. Theory Tech.*, vol. 54, no. 4, pp. 1340–1349, Jun. 2006.

[19] H. Kang, Y. Cho, and D. Youn, "On compensating nonlinear distortions of an OFDM system using an efficient adaptive predistorter," *IEEE Trans. Commun.*, vol. 47, no. 4, pp. 522–526, Apr. 1999.

[20] T. Wang and J. Ilow, "Compensation of nonlinear distortions with memory effects in OFDM transmitters," in *Proc. IEEE Global Telecommun. Conf.*, 2004, pp. 2398–2403.

[21] A. Sano and L. Sun, "Identification of hammerstein-wiener system with application to compensation for nonlinear distortion," in *Proc. 41st SICE Annu. Conf.*, 2002, pp. 1521–1526.

[22] M. Cheong, S. Werner, M. Bruno, J. Figueroa, J. Cousseau, and R. Wichman, "Adaptive piecewise linear predistorters for nonlinear power amplifiers with memory," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, no. 7, pp. 1519–1532, Jul. 2012.

[23] D. Bharadia, E. McMilin, and S. Katti, "Full Duplex Radios," in *Proc. ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, 2013, pp. 375–386.

[24] L. Anttila, D. Korpi, V. Syrjälä, and M. Valkama, "Cancellation of power amplifier induced nonlinear self-interference in full-duplex transceivers," in *Proc. 47th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2013, pp. 1193–1198.

[25] D. Korpi, Y.-S. Choi, T. Huusari, S. Anttila, L. Talwar, and M. Valkama, "Adaptive nonlinear digital self-interference cancellation for mobile in-band full-duplex radio: Algorithms and RF measurements," in *Proc. IEEE Global Commun. Conf.*, Dec. 2015, pp. 1–7.
[26] K. Komatsu, Y. Miyaji, and H. Uehara, "Frequency-domain hammerstein self-interference canceller for in-band full-duplex OFDM systems," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2017, pp. 1–6.
[27] P. P. Campo *et al.*, "Gradient-adaptive spline-interpolated lut methods for low-complexity digital predistortion," *IEEE Trans. Circuits Syst. I, Reg. Papers*, to be published, doi: 10.1109/TCSI.2020.3034825.
[28] M. Scarpiniti, D. Comminiello, R. Parisi, and A. Uncini, "Nonlinear spline adaptive filtering," *Signal Process.*, vol. 93, no. 4, pp. 772–783, 2013.
[29] M. Scarpiniti, D. Comminiello, R. Parisi, and A. Uncini, "Hammerstein uniform cubic spline adaptive filters: Learning and convergence properties," *Signal Process.*, vol. 100, pp. 112–123, 2014.
[30] M. Scarpiniti, D. Comminiello, R. Parisi, and A. Uncini, "Novel cascade spline architectures for the identification of nonlinear systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 7, pp. 1825–1835, Jul. 2015.
[31] C. De Boor, *Practical Guide to Splines*. New York, NY, USA: Springer-Verlag, 1978.
[32] M. Vetterli, J. Kovaçević, and V. K. Goyal, *Foundations of Signal Processing*. Cambridge, U.K.: Cambridge Univ. Press, 2014.
[33] H. Prautzsch, W. Boehm, and M. Paluszny, *Bèzier and B-Spline Techniques*. Berlin, Germany: Springer, 2013.
[34] N. Safari, N. Holte, and T. Roste, "Digital predistortion of power amplifiers based on spline approximations of the amplifier characteristics," in *Proc. IEEE 66th Veh. Technol. Conf.*, Sep. 2007, pp. 2075–2080.
[35] J. K. Cavers, "Amplifier linearization using a digital predistorter with fast adaptation and low memory requirements," *IEEE Trans. Veh. Technol.*, vol. 39, no. 4, pp. 374–382, Nov. 1990.
[36] M. O'Droma, S. Meza, and Y. Lei, "New modified saleh models for memoryless nonlinear power amplifier behavioural modelling," *IEEE Commun. Lett.*, vol. 13, no. 6, pp. 399–401, Jun. 2009.
[37] E. Hänsler and G. Schmidt, *Acoustic Echo and Noise Control: A Practical Approach*. Hoboken, NJ, USA: Wiley, 2005, vol. 40, pp. 43–46.
[38] P. S. R. Diniz *et al.*, "Adaptive Filtering: Algorithms and Practical Implementation," *The International Series in Engineering and Computer Science*, pp. 147–150, 2008.
[39] D. P. Mandic and V. Su LeeGoh, *Complex Valued Nonlinear Adaptive Filters: Noncircularity, Widely Linear and Neural Models*. Hoboken, NJ, USA: Wiley, 2009.
[40] D. Korpi *et al.*, "Full-duplex mobile device: Pushing the limits," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 80–87, Sep. 2016.
[41] D. Korpi, Y. Choi, T. Huusari, L. Anttila, S. Talwar, and M. Valkama, "Adaptive nonlinear digital self-interference cancellation for mobile in-band full-duplex radio: Algorithms and RF measurements," in *Proc. 2015 Global Commun. Conf.*, 2015, pp. 1–7.
[42] C. B. Barneto *et al.*, "Full-duplex OFDM radar with LTE and 5 G NR waveforms: Challenges, solutions, and measurements," *IEEE Trans. Microw. Theory Tech.*, vol. 67, no. 10, pp. 4042–4054, Oct. 2019.
[43] D. Korpi, M. Heino, C. Icheln, K. Haneda, and M. Valkama, "Compact inband full-duplex relays with beyond 100 dB self-interference suppression: Enabling techniques and field measurements," *IEEE Trans. Antennas Propag.*, vol. 65, pp. 960–965, Feb. 2017.
[44] M. Abdelaziz, L. Anttila, A. Brihuega, F. Tufvesson, and M. Valkama, "Digital predistortion for hybrid MIMO transmitters," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 3, pp. 445–454, Jun. 2018.
[45] 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.
[46] P. P. Campo, V. Lampu, A. Meirhaeghe, J. Boutellier, L. Anttila, and M. Valkama, "Digital predistortion for 5G small cell: GPU implementation and RF measurements," *J. Signal Process. Syst.*, pp. 1–12, 2019.
[47] C. Fager, T. Eriksson, F. Barradas, K. Hausmair, T. Cunha, and J. C. Pedro, "Linearity and efficiency in 5G transmitters: New techniques for analyzing efficiency, linearity, and linearization in a 5G active antenna transmitter context," *IEEE Microw. Mag.*, vol. 20, no. 5, pp. 35–49, May 2019.
[48] 3GPP Tech. Spec. 38.141-2, NR; Base Station (BS) Conformance Testing, Part 2, v15.1.0 (*Release 15*), Mar. 2019.

Pablo Pascual Campo (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees in telecommunications and electrical engineering in 2016 and 2018, respectively, from Universidad Politécnica de Madrid, Madrid, Spain. He is currently working toward the D.Sc. degree with Tampere University, Department of Electrical Engineering, Tampere, Finland. His research interests include digital predistortion, full-duplex systems and applications, and signal processing for wireless communications at the mmW bands.

Lauri Anttila (Member, IEEE) received the D.Sc. (Tech.) degree (with distinction) in 2011 from Tampere University of Technology (TUT), Finland. Since 2016, he has been a University Researcher with the Department of Electrical Engineering, Tampere University (formerly TUT). In 2016–2017, he was a Visiting Research Fellow with Aalto University, Finland. His research interests include radio communications and signal processing, with a focus on the radio implementation challenges in systems such as 5G, full-duplex radio, and large-scale antenna systems.

Dani Korpi received the D.Sc. degree (Hons.) in electrical engineering from the Tampere University of Technology (TUT), Finland, in 2017. He is currently a Senior Specialist with Nokia Bell Labs in Espoo, Finland. He received the Best Dissertation Award of the year in TUT, as well as the Finnish Technical Sector's Award for the Best Doctoral Dissertation of 2017. His research interests include inband full-duplex radios, machine learning for wireless communications, and beyond 5G radio systems.

Mikko Valkama (Senior Member, IEEE) received the D.Sc. (Tech.) degree (with honors) from the Tampere University of Technology (TUT), Finland, in 2001. In 2003, he was a Visiting Postdoc Research Fellow with SDSU, San Diego, CA. He is currently a Full Professor and Department Head of Electrical Engineering with the newly formed Tampere University (TAU), Finland. His research interests include radio communications, radio localization, and radio-based sensing, with emphasis on 5G and beyond mobile radio networks.