Digital predistortion for 5G small cell : GPU implementation and RF measurements

Journal of Signal Processing Systems
https://doi.org/10.1007/s11265-019-01502-4

Digital Predistortion for 5G Small Cell: GPU Implementation and RF Measurements

Pablo Pascual Campo[1] · **Vesa Lampu**[1] · **Alexandre Meirhaeghe**[2] · **Jani Boutellier**[2,3] · **Lauri Anttila**[1] · **Mikko Valkama**[1]

Received: 20 Jun 2019 / Revised: 25 Sep 2019 / Accepted: 15 Nov 2019

Abstract

In this paper, we present a high data rate implementation of a digital predistortion (DPD) algorithm on a modern mobile multicore CPU containing an on-chip GPU. The proposed implementation is capable of running in real-time, thanks to the execution of the predistortion stage inside the GPU, and the execution of the learning stage on a separate CPU core. This configuration, combined with the low complexity DPD design, allows for more than 400 Msamples/s sample rates. This is sufficient for satisfying 5G new radio (NR) base station radio transmission specifications in the sub-6 GHz bands, where signal bandwidths up to 100 MHz are specified. The linearization performance is validated with RF measurements on two base station power amplifiers at 3.7 GHz, showing that the 5G NR downlink emission requirements are satisfied.

Keywords Digital predistortion (DPD) · 5G · GPU · Real-time · High data rate

1 Introduction

Modern wireless communications are continuously evolving to offer higher data rates to an ever-increasing number of users. Owing to this, a high spectral efficiency is particularly sought within the scarce radio frequency spectrum [9]. In this context, higher-order symbol alphabets as well as more wideband signals are beginning to be utilised. One recent example is the fifth generation (5G) new radio (NR) standard [1], where the physical layer is based on orthogonal frequency division multiplexing (OFDM). The 5G NR standard specifies signal bandwidths up to 100 MHz in frequency range (FR) 1 (sub-6 GHz), and up to 400 MHz in FR2 (24-53 GHz).

These characteristics allow the increase of data rates in the wireless transmission to a new level, but the transmit (TX) chain requirements also become naturally tougher, since the transmitters' nonlinearities are excited more aggressively. The main challenge in this context is the power amplifier (PA), which is typically operated close to saturation to ensure a high power efficiency [20]. At this point, we can distinguish two distinct effects that are particularly harmful for the PA: the high peak-to-average power ratio (PAPR) of the transmit signal due to the random constructive addition of different parallel subcarriers, and the wide bandwidth of the signal, which excite the memory effects present in the PA [2].

To avoid these undesired effects in the TX chain, a backoff can be applied to the output power of the PA, thus driving it in a more linear region, at the expense of a reduction in the power efficiency, since it decreases as the output power deviates from the 1 dB compression point [5]. Another more elegant solution is to make use of the so-called digital predistortion (DPD). Digital predistortion [4, 14, 22, 25, 31] is a PA linearization technique that estimates

✉ Pablo Pascual Campo
pablo.pascualcampo@tuni.fi

Vesa Lampu
vesa.lampu@tuni.fi

Lauri Anttila
lauri.anttila@tuni.fi

[1] Department of Electrical Engineering, Tampere University, Tampere, Finland

[2] Department of Computing Sciences, Tampere University, Tampere, Finland

[3] Department of Computer Science, University of Vaasa, Vaasa, Finland

the PA behavior and tries to invert it in a preceding stage. Hence, the undesired nonlinear and memory contributions at the PA output are minimized, while a high output power is still reached and the power efficiency remains good. There are many DPD models studied in the literature, most of them being simplifications of the Volterra series, such as the Volterra series with dynamic deviation reduction [10], the generalized memory polynomial (GMP) [23, 24], and the memory polynomial (MP) [8].

Literature has examples of such DPD algorithms being implemented on FPGAs, such as [7, 17, 21, 28]. While these works demonstrate favorable linearization performances with real-time computing, the effort required by FPGA programming is high. This is due to FPGA requiring compile and synthesis for each code adjustment, and these are time consuming tasks. As a consequence, in this paper the high throughput computation is achieved with a GPU. Previous works have shown that GPU implementations can achieve similar data rate performances as the FPGA equivalents, with the added benefit of design flexibility [18].

The software based DPD solution presented in this paper provides real-time predistortion performance on an off-the-shelf mobile CPU that has an on-chip GPU inside and a modest TDP (Thermal Design Power) of 15 W. Such processors are available e.g. inside fanless industrial PCs and laptops [19]. On one hand, real-time filtering is achieved by the high parallelism offered by the GPU, and on the other hand by performing pipelined data transfer between the CPU and GPU. The parallel processing offered by the multicore CPU allows executing DPD coefficient learning concurrently with filtering. The software implementation is based on a dataflow programming environment [6] that takes care of data transfer and synchronization between the CPU cores and the GPU. The GPU code is written in OpenCL for cross-platform portability, whereas the DPD functionalities executed on the CPU cores are written in C.

To support the proposed real-time implementation, two independent experiments with two real PA measurement setups have been conducted. To generate a tough nonlinearity in the TX chain, we have excited the PAs with a transmit signal of 100 MHz bandwidth and high PAPR. The PAs are also operated close to their saturation points to achieve good and realistic power efficiencies. The first measured PA is designed to perform as the amplification stage of a band 78 local area (i.e. small cell) BS, with a transmit power below +24 dBm. The second PA corresponds to a medium range BS, with a transmit power below +38 dBm, in order to show that the proposed technique is also capable of linearizing higher power PAs. In this work, we have used the error vector magnitude (EVM) and adjacent channel power ratio (ACPR) metrics to test the in-band and out-of-band linearization performances, respectively. In both cases, the nonlinear and memory distortions are successfully suppressed, satisfying the NR specifications and showing that the proposed implementation is capable of linearizing the target PAs in real-time.

The rest of the paper is organized as follows. Section 2 presents a mathematical description of the algorithm modeling the nonlinear and memory effects of an arbitrary PA. Stemming from this modeling, the implementation in the GPU is carried out. Section 3 details the GPU implementation, having special emphasis on the asynchronous GPU data transfer and DPD filtering. Section 4 introduces the EVM and ACPR performance metrics, and follows with a description of the RF measurement setup used to conduct the two experiments. The tested PAs and their specific characteristics are also presented. Section 5 presents the conducted RF measurements, incorporating the local area BS and medium range BS target PAs. For each case, we show the PA nonlinear behavior with and without the proposed linearization technique. Finally, Section 6 summarizes the main findings of this work.

Mathematical Notation Throughout this paper, we adopt complex-valued baseband modeling of the system. Matrices are represented by capital boldface letters, e.g., $\mathbf{A} \in \mathbb{C}^{(M \times N)}$. Vectors are represented with lowercase boldface letters, e.g., $\mathbf{v} \in \mathbb{C}^{M \times 1} = [v_1 \; v_2 \; \cdots \; v_M]^T$. Ordinary transpose, conjugate-transpose, and conjugation operators are denoted by $(\cdot)^T$, $(\cdot)^H$, and $(\cdot)^*$, respectively. Additionally, the absolute value and ceil operators are represented as $|\cdot|$ and $\lceil \cdot \rceil$, respectively.

2 Digital Predistortion Algorithm Description

In this section, the decorrelation-based DPD algorithm from [3], which is utilized in this work, is described. Both the main and learning paths are detailed, where the predistortion and coefficient estimation are carried out, respectively. The MP model is adopted as the behavioral identification model for this work, as it has been proven to obtain high performance when modeling the nonlinear behavior of real PAs in different scenarios [3, 15, 30]. However, we note that the learning solution described in [3] supports any other Volterra-based model as well.

The complete scheme of the closed-loop DPD architecture adopted in this work is presented in Fig. 1, where the predistortion and learning stages are depicted. In this context, $x[n]$ is the original baseband signal, $x_{\mathrm{DPD}}[n]$ is the predistorted signal, and $y_{\mathrm{RF}}(t)$ is the RF PA output signal, to be finally transmitted with low levels of nonlinear distortion thanks to the DPD operation.

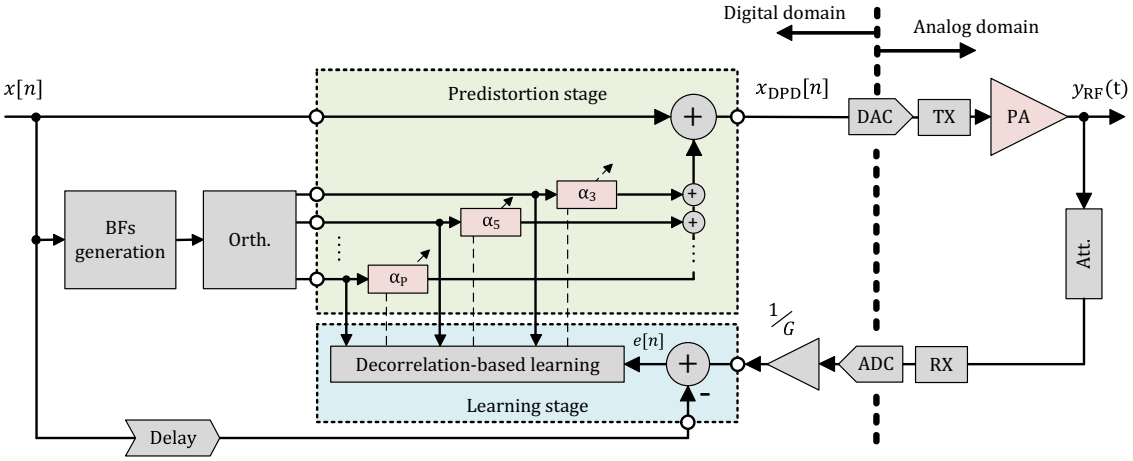


Figure 1 An illustration of the closed-loop decorrelation-based DPD scheme used in this work, showing both predistortion and learning stages. The $\frac{1}{G}$ factor (G being the estimated gain of the measurement loop) is obtained with a linear fitting LS technique in the digital domain.

Following this nomenclature, and considering a classical MP PA model, we can express the complex envelope model of $y_{\mathrm{RF}}(t)$ as

$$y_{\mathrm{MP}}[n] = \sum_{\substack{p=1 \\ p\,\mathrm{odd}}}^{P} \sum_{m=0}^{M} a_{p,m}\, x[n-m]|x[n-m]|^{p-1}, \tag{1}$$

where P is the polynomial order and M the memory depth of the MP model, and $a_{p,m}$ contains the PA model coefficients. This expression constitutes the base to formulate the proposed decorrelation based DPD structure in the following subsection.

2.1 DPD Processing

The main idea of the DPD algorithm is to inject weighted nonlinear basis functions (BF), similar to those in the assumed PA model in (1), to the baseband signal, such that the nonlinear distortion at the PA output is minimized. Following this principle, the predistorter operation, as depicted in Fig. 1, is formulated as

$$x_{\mathrm{DPD}}[n] = x[n] + \sum_{\substack{p=1 \\ \mathrm{p\,odd}}}^{P} \sum_{m=0}^{M} \alpha^*_{p,m} x[n-m]|x[n-m]|^{p-1}, \tag{2}$$

where $\alpha_{p,m}$ are the coefficients of the model. Note that this double summation also includes an overlapping linear term $\alpha^*_{1,0}\, x[n]$, when $p = 1$ and $m = 0$, however it is omitted in the DPD processing.

Naturally, this expression can be formulated with matrix notations when processing over a signal block of length N, as

$$\mathbf{x}_{\mathrm{DPD}} = \mathbf{x} + \mathbf{X}^{[2:\mathrm{end}]}_{\mathrm{BF,Q}}\, \boldsymbol{\alpha}^*, \tag{3}$$

where $\mathbf{x} \in \mathbb{C}^{(N\times 1)} = [x[n]\ x[n+1]\ \cdots\ x[n+N-1]]^T$, $\boldsymbol{\alpha} \in \mathbb{C}^{(\lceil \frac{P}{2} \rceil (1+M)-1 \times 1)}$ contains the estimated coefficients of the DPD model, and the matrix $\mathbf{X}_{\mathrm{BF,Q}} \in \mathbb{C}^{(N \times \lceil \frac{P}{2} \rceil (1+M))}$ is the set of orthogonalized basis functions of the transmit signal, in this expression only containing the columns from the second to the last. The orthogonalized BF matrix is defined as

$$\mathbf{X}_{\mathrm{BF,Q}} = \mathbf{X}_{\mathrm{BF}}\mathbf{Q}, \tag{4}$$

where $\mathbf{X}_{\mathrm{BF}} \in \mathbb{C}^{(N \times \lceil \frac{P}{2} \rceil (1+M))}$ is the original, monomial, basis function matrix defined as

$$\mathbf{X}_{\mathrm{BF}} = \left[\, \mathbf{x}_{1,0} \cdots \mathbf{x}_{P,0}\ \mathbf{x}_{1,1} \cdots \mathbf{x}_{P,1} \cdots \mathbf{x}_{1,M} \cdots \mathbf{x}_{P,M} \,\right], \tag{5}$$

with

$$\mathbf{x}_{p,m} = \begin{bmatrix} \mathbf{0}_m \\ x[n]|x[n]|^{p-1} \\ x[n+1]|x[n+1]|^{p-1} \\ \vdots \\ x[n+N-1-m]|x[n+N-1-m]|^{p-1} \end{bmatrix}, \tag{6}$$

and where $\mathbf{0}_m$ is a vector of m zeros. The matrix $\mathbf{Q} \in \mathbb{C}^{(\lceil \frac{P}{2} \rceil (1+M) \times \lceil \frac{P}{2} \rceil (1+M))}$ is an upper triangular orthogonalization matrix with a one as its first element to leave the first column of $\mathbf{X}_{\mathrm{BF}}$ (i.e., $\mathbf{x}_{1,0} = \mathbf{x}$) untouched, and it is formally defined as

$$\mathbf{Q} = \begin{bmatrix} 1 & q_{1,2} & q_{1,3} & \cdots & q_{1,\lceil \frac{P}{2} \rceil (1+M)} \\ 0 & q_{2,2} & q_{2,3} & \cdots & q_{2,\lceil \frac{P}{2} \rceil (1+M)} \\ 0 & 0 & q_{3,3} & \cdots & q_{3,\lceil \frac{P}{2} \rceil (1+M)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & q_{\lceil \frac{P}{2} \rceil (1+M),\lceil \frac{P}{2} \rceil (1+M)} \end{bmatrix}. \tag{7}$$

The orthogonalization is necessary because, in general, the original BFs of the form $x[n]|x[n]|^{p-1}$, $p = 1, 3, 5, \ldots$ are

highly mutually correlated, and this will slow down the convergence speed in the coefficient learning process [11, 16]. In the proposed solution, a Cholesky decomposition-based orthogonalization method is adopted to orthogonalize the basis functions, as outlined for example in [16]. The orthogonalization matrix $\mathbf{Q}$ can be precomputed based on the statistics of the input signal $x[n]$, and thus its computation does not affect the real-time computing load of the DPD system.

As a final item, the DPD main path processing complexity of the predistorter is analyzed in terms of floating operations (FLOPs) per linearized sample. As similarly presented in [3], the DPD main path computational effort can be separated in two stages:

- BF generation $\rightarrow 3\lceil\frac{P}{2}\rceil - 1$ FLOPs,
- Main path processing $\rightarrow 8\lceil\frac{P}{2}\rceil(M+1) - 2$ FLOPs.

2.2 DPD Learning

In this subsection, we focus on the iterative estimation of the coefficient vector $\boldsymbol{\alpha}$, to formulate an efficient and computationally simple DPD learning solution. The learning of $\boldsymbol{\alpha}$ is based on the decorrelation principle, where the correlation between the nonlinear distortion at the PA output and the BFs representing the nonlinear distortion is minimized.

Firstly, the linear signal component (i.e. $x[n]$) needs to be subtracted from the observed PA output signal, hence leaving only the distortion terms. Note that an estimate of the complex linear gain of the measurement loop is required, which can be easily obtained by utilizing some linear estimation technique, such as least squares (LS). Hence, the instantaneous error signal $e[n]$ at time instant n can be formulated as

$$e[n] = \frac{y[n]}{G} - x[n], \tag{8}$$

where $y[n]$ is the complex envelope of the measured PA output signal, and G represents the estimated complex gain of the measurement loop, which in this case is estimated with linear LS fitting.

The learning principle is based on finding the DPD coefficients that minimize the correlation between the nonlinearities at the PA output and the generated BFs. For this purpose, a learning solution based on a least mean square (LMS) algorithm [26, 27] is used. The block-adaptive decorrelation-based DPD coefficient update at iteration i is now formulated as [3]

$$\boldsymbol{\alpha}_{i+1} = \boldsymbol{\alpha}_i - \mu_\alpha \left(\mathbf{e}^H \mathbf{X}_{\mathrm{BF,Q}}^{[2:\mathrm{end}]}\right)^T, \tag{9}$$

where $\mathbf{e} \in \mathbb{C}^{(N\times 1)} = [e[n]\ e[n+1]\ \cdots\ e[n+N-1]]^T$ is the error signal vector, $\mathbf{X}_{\mathrm{BF,Q}}$ is the set of orthogonalized BFs defined in (4), and μ_α is the learning rate of the LMS algorithm.

This coefficient vector is then used in the predistortion stage, within the N-size data block, to compensate for the static nonlinearity of the PA. The process is then iterated until convergence of $\boldsymbol{\alpha}_i$ is reached.

The learning stage processing complexity, following the block-adaptive filtering of size N used in the update, in terms of FLOPs per N samples can be expressed as similarly done in [3]:

- Gain estimation and scaling $(G) \rightarrow 18N - 4$ FLOPs,
- DPD estimation $\rightarrow 2\lceil\frac{P}{2}\rceil(M+1)(4N+1)$ FLOPs.

3 DPD Implementation

The decorrelation-based DPD was implemented targeting a mobile Intel CPU that has an integrated GPU. Real-time implementation needed to consider the fact that DPD filtering has to run in real-time more or less continuously, whereas the learning functionality can operate in the background updating the DPD weights whenever required. To this extent, the performance-critical filtering operation was mapped to the integrated GPU, whereas learning was left to be executed on a CPU core.

The targeted sample rate of the filtering operation was set to 400 Msamples/s, which resulted in a final data rate of 3.2 Gbytes/s considering that the numeric precision was complex 32-bit float. Looking at Fig. 1, it can be seen that the basis function generation (BF generation) and orthogonalization (Orth.) are on the critical real-time filtering path. Moreover, orthogonalization yields $P-1$ signals, each having the data rate of 3.2 Gbytes/s. For $P = 9$ this means that, including the main signal path, the data bandwidth between orthogonalization and filtering is 28.8 Gbytes/s. Such a data rate is not feasible for inter-circuit communication, and for this reason the only viable choice was to perform the BF generation, orthogonalization and DPD filtering within one single GPU kernel (function). Consequently, the input and output data rates of the GPU remained at 3.2 Gbytes/s. The only drawback of this function mapping was that the BF generation and orthogonalization had to be performed redundantly in the CPU-mapped DPD coefficient learning block. This has some negative impact on the maximum coefficient update frequency, as well as on power dissipation.

Figure 2 shows a Gantt chart of the real-time processing of DPD filtering on the GPU for three sample blocks, $j = 0, 1, 2$, each block consisting of $2^{18} - 1$ samples. The processing schedule has been acquired from the GPU programming API (Application Programming Interface) and reflects real start and end times for each operation. The

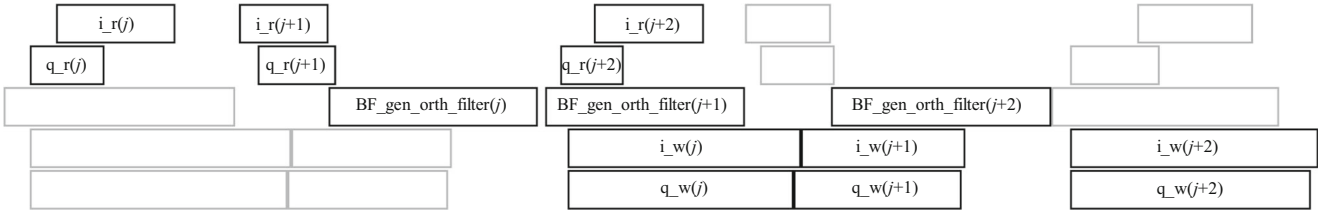


Figure 2 Asynchronous GPU data transfer and DPD filtering of three sample blocks, $j = 0, 1, 2$.

Gantt chart has five lines, which are in top-down order as stated below

1. Transfer of I samples to the GPU (i_r)
2. Transfer of Q samples to the GPU (q_r)
3. Joint BF generation, orthogonalization and filtering of I and Q samples (BF_gen_orth_filter)
4. Transfer of I samples from the GPU to the CPU (i_w)
5. Transfer of Q samples from the GPU to the CPU (q_w)

Therefore, the Gantt chart shows that

- Data transfer to/from the GPU necessarily needs to happen in parallel with filtering to achieve the desired sample rate.
- I and Q sample streams need to be transferred in parallel for maximum performance.
- Both data read from the GPU and the filtering consume a similar amount of time and hence together form the performance limit.

The software description of the DPD was authored within a *dataflow computing* framework [6] that provided support for multiprocessing, GPU interfacing and synchronization between CPU cores and the GPU. The DPD filtering function that was executed on the GPU was written in the OpenCL language[29], which allows the filtering to be executed on almost any modern GPU. The other functionalities of the DPD were written in C language and executed on the CPU cores.

The GPU-mapped part of the code was parallelized in a straightforward way such that each complex input sample is computed in a separate OpenCL *work item*. Profiling of several GPU code versions revealed that the best performance is achieved when orthogonalization and filtering are computed jointly in a nested loop. This fastest, yet redundant GPU code requires 448 FLOPs per complex input sample, whereas the non-redundant version requires 322 FLOPs, but performs slightly worse.

4 Experimental Verification

For the purposes of verifying the operation of the implemented DPD algorithm, series of tests were conducted. First, the real-time capabilities of the algorithm were studied by data rate performance testing. These tests were carried out on two different CPUs with integrated GPUs. Additionally, the linearization performance of the algorithm was tested with two PAs. In order to validate the performance, the EVM and ACPR are used as figures-of-merit. The EVM can be defined as

$$\mathrm{EVM} = \sqrt{\frac{\sum_{k=1}^{K} |s_{\mathrm{meas}}(k) - s(k)|^2}{\sum_{k=1}^{K} |s(k)|^2}} \times 100\ (\%), \tag{10}$$

where $s(k)$ are the original data symbols, $s_{\mathrm{meas}}(k)$ are the measured symbols which have been equalized for any linear distortion, and K is the total number of symbols over which the EVM is calculated. Likewise, the ACPR can be determined as

$$\mathrm{ACPR} = 10 \log_{10}\left(\frac{P_{\mathrm{main}}}{P_{\mathrm{adj}}}\right)\ (\mathrm{dB}), \tag{11}$$

where P_{main} and P_{adj} are the measured powers on the main and the adjacent channel of the PA output signal. In this paper, the ACPR is given separately for the left and right adjacent channels.

As a last comment, Table 1 gathers the BS configurations, requirements, and limits specified in the 3GPP's NR BS radio transmission and reception standard [1].

Table 1 BS configurations, limits, and requirements according to the new radio (NR) 3GPP standard [1]. There is no upper limit for the radiated power of the Wide Area BS.

Type of BS	TX power	EVM limit (64-QAM)	Absolute basic limit
Local Area	$\leq$24 (dBm)	8%	−32(dBm/MHz)
Medium Range	$\leq$38 (dBm)	8%	−25 (dBm/MHz)
Cat. A Wide Area	–	8%	−15 (dBm/MHz)
Cat. B Wide Area	–	8%	−13 (dBm/MHz)

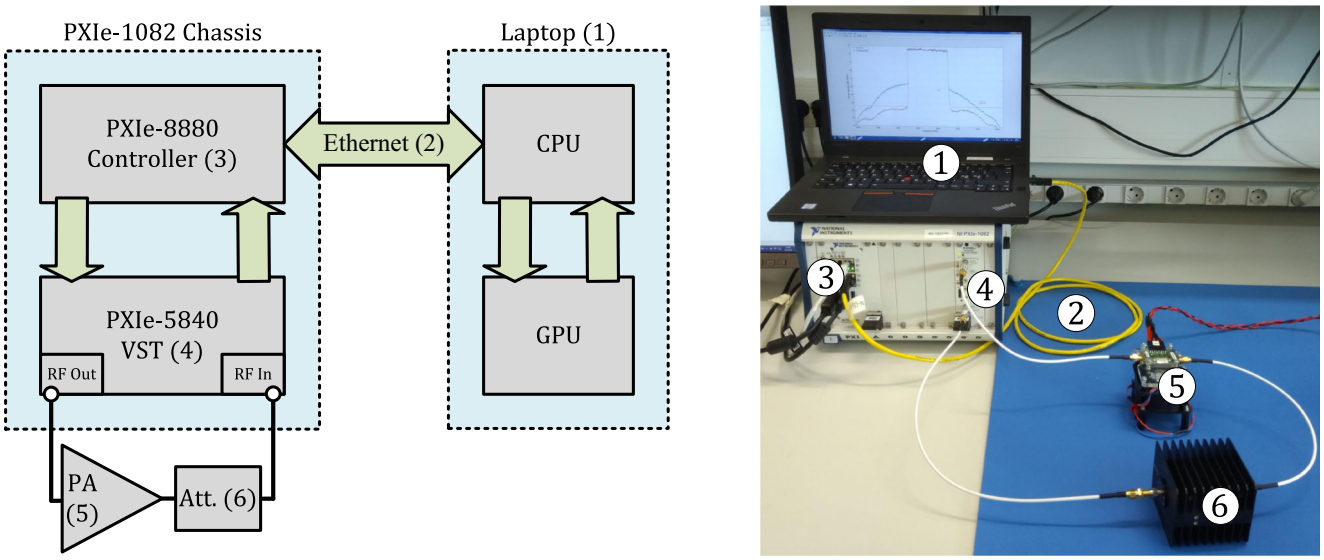


Figure 3 Experimental DPD setup used in the RF measurements.

Measurement Setup The linearization performance of the implemented DPD algorithm was measured with the setup shown in Fig. 3. The setup includes a laptop, which has the Intel CPU incorporated, a National Instruments (NI) PXIe-1082 chassis holding an NI PXIe-8880 embedded controller, an NI PXIe-5840 vector signal transceiver (VST), and the PA under test, connected to an attenuator. The laptop is connected to the embedded controller via an Ethernet cable. For the measurements, two PAs were utilized: a Skyworks SKY66293-21 and an Analog Devices HMC1114PM5E gallium nitride (GaN) PA. Both of the PAs are featured in Fig. 4. The PAs were measured at 3.7 GHz, where the SKY66293-21 typically provides 32.6 dB of gain while the HMC1114PM5E provides approximately 33.8 dB of gain. Due to their available output powers, the former PA is aimed towards local area BS and the latter towards medium range BS according to the new 3GPP NR standards featured in Table 1. For the purposes of the measurements, the output power of the SKY66293-21 was tuned to +22 dBm and the output power of HMC1114PM5E to +36 dBm, which satisfy the local and medium range BS TX power limits, respectively. At the same time, the transmit powers are close to their respective upper limits, thus corresponding to realistic, nonlinear, operation points.

The transmit signal is generated on the laptop computer, and the signal is stored in a file for future use. The signal used in the measurements is an NR FR1 compliant OFDM signal with 100 MHz bandwidth, 64-QAM subcarrier modulation, and 30 kHz subcarrier spacing. The native minimum sampling rate for this configuration is 122.88 MHz, but for DPD processing an oversampling factor of three is used, thus yielding a sampling rate of 368.64 MHz. The signal generator also includes PAPR reduction based on iterative clipping and filtering (ICF), and windowing to reduce the spectral sidelobes of the OFDM signal. After PAPR reduction, the generated signal has a PAPR of 7 dB. The ICF technique also creates in-band distortion, such that an EVM floor of about 4.8% is induced.

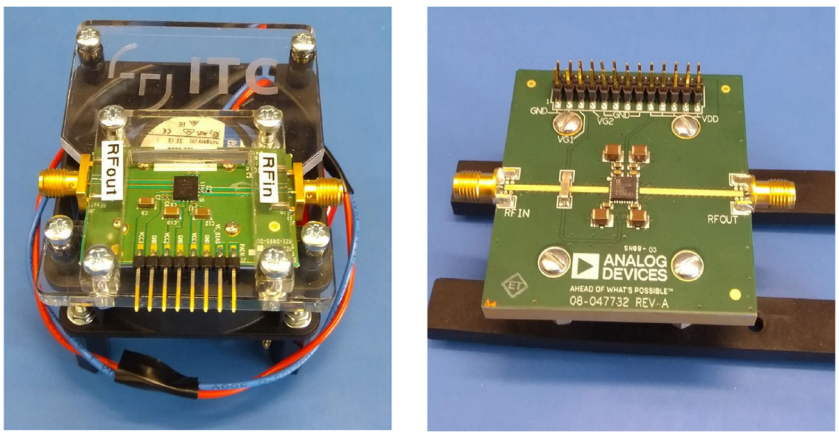


Figure 4 The PAs used in the measurements. The figure on the left corresponds to the Skyworks SKY66293-21, and the figure on the right shows the Analog Devices HMC1114PM5E.

Table 2 Computation platforms used for the DPD performance results. When presenting the number of cores, the first number refers to the physical cores, and the number in the parenthesis to the virtual cores.

Tag	CPU	GPU	OS/Compiler
HD	Intel Core i7-6700HQ 2.6 GHz, 4(8) cores	Intel HD Graphics 530 OpenCL 2.0, driver r3.0.57406	Ubuntu 18.04 g++ 7.3.0
NEO	Intel Core i7-8650U 1.9 HGz, 4(8) cores	Intel UHD Graphics 620 OpenCL 2.0, driver 19.17.12918	Ubuntu 18.04 g++ 7.4.0

After generating the baseband signal, the measurement procedure is as follows. A block of the signal with length $N = 2^{16} - 1$ is sent to the embedded controller via a TCP/IP connection, and then further forwarded to the VST for transmission. The VST transmits the signal through the PA and attenuator and finally receives it on the receiver side of the RF front-end. The received signal is sent to the embedded controller, which synchronizes the transmit and received signals. For the purpose of improving the SNR in the DPD estimation, the same block of signal is transmitted five times, and the received signals are averaged. The averaging is also computed on the embedded controller. After the averaging is done, the signal is sent back to the laptop via the TCP/IP connection. The laptop CPU saves the received data on a file. The transmit and received signals are then used to determine the weights on the CPU, and the next block of signal is predistorted on the GPU using the newly calculated weights. The predistorted signal is then sent to the embedded controller for transmission. This loop is run until the weights have converged.

5 Performance Results

In this section, we present the performance of the proposed real-time DPD implementation. Specifically, we study the achieved data rate with the CPU processors and the nonlinearity suppression in terms of EVM and ACPR.

5.1 Data Rate Performance

The computation performance was benchmarked on two mobile processors that have built-in GPUs. The first processor was a 2015 Intel quad-core processor ('Skylake') with 45 W TDP, and the second one a 2017 Intel quad-core ('Kaby Lake R') with TDP of 15 W. The details of the platforms around these processors are listed in Table 2.

Table 3 shows the learning and filtering performance of the DPD with various transmission block sizes. It can be seen that with the GPU of the HD platform the filtering performance constantly increases with the block size, approaching asymptotically 400 Msamples/s. With the more recent GPU of the NEO platform, 400 Msamples/s are achieved already with a block size of $2^{17} - 1$ samples, and increasing the block size beyond $2^{18} - 1$ actually reduces the achievable sample rate.

The GPU-mapped functionality of the DPD needs to compute 448 FLOPs for each complex input sample to perform the filtering. On the HD platform, for block size $2^{19}-1$ this equals to 180 GFLOPs/s. As the Intel product specification [12] reveals that the maximum operating frequency of the HD GPU is 1.050 GHz, and that the HD GPU can theoretically provide a maximum of 384 FLOPs per clock cycle [13], we can calculate that the GPU achieves 45% of its theoretical floating point performance. Acknowledging that the GPU also needs to perform memory reads and writes, as well as some integer arithmetic-logical operations, it can be concluded that the GPU is well-utilized.

The learning functionality is executed and profiled on a single CPU core of the HD and NEO platforms, and shows considerably smaller performance than filtering. In practice this is not a problem, since the learning functionality can be executed as a background process whenever needed. Furthermore, to ensure high sample rate of the system, the learning can also be executed for shorter block lengths than those of the filtering.

Table 4 shows the achieved filtering sample rates in the context of an earlier work. This previous work explored

Table 3 Columns 2-4: performance in terms of Msamples/second of the DPD filtering and learning stages; column 5: filtering latency in milliseconds.

Function	Block size (samples)	HD (Ms/s)	NEO (Ms/s)	Latency (ms)
Filtering (GPU)	10 000	86.8	103.3	0.10
	32 767	203.0	236.5	0.14
	65 535	280.8	328.7	0.20
	131 071	324.0	432.5	0.30
	262 143	392.8	483.6	0.54
	524 287	401.7	400.3	1.31
Learning (CPU)	–	5.2	6.2	–

Table 4 Comparison between the peak data rates of earlier and this work.

Processing unit	Peak throughput (Msamples/s)
Kepler GPU @ 852 MHz [18]	153.5
Maxwell GPU @ 998 MHz [18]	221.8
ARMv7 CPU @ 2.32 GHz [18]	214.4
ARMv8 CPU @ 1.91 GHz [18]	200.1
HD 530 GPU (HD) @ 1.05 GHz	401.7
UHD 620 GPU (NEO) @ 1.15 GHz	483.6

Table 5 Linearization performance of the tested PAs, in terms of EVM and ACPR performance metrics. The PAPR of the original signal with no PAPR reduction method is 10 dB, leading to the values of 4.8 % and 52.07 dB of original EVM and ACPR, respectively.

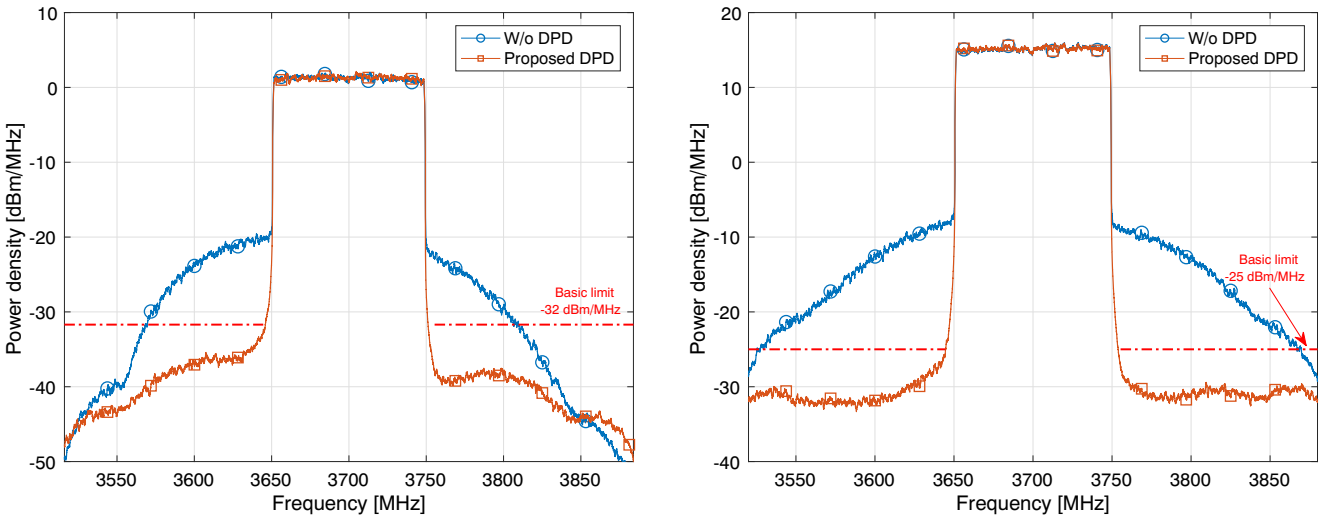
PA model	Type of measurement	EVM (%)	ACPR (L/R) (dB)
SKY66293-21	Without DPD	8.85	−24.80 / −28.70
	With DPD	5.60	−40.00 / −40.90
HMC1114PM5E	Without DPD	11.31	−26.30 / −26.30
	With DPD	6.16	−38.60 / −37.80

both the suitability of a GPU and a SIMD-enabled CPU for filtering, whereas this work concentrates only on GPUs for filtering purposes. It can be seen from the table that the demonstrated sample rates are around twofold compared to the previous work, acknowledging that in this work the DPD has 10 branches and memory depth 1, whereas in [18] the DPD had 5 branches and a memory depth of 5.

5.2 Linearization Performance

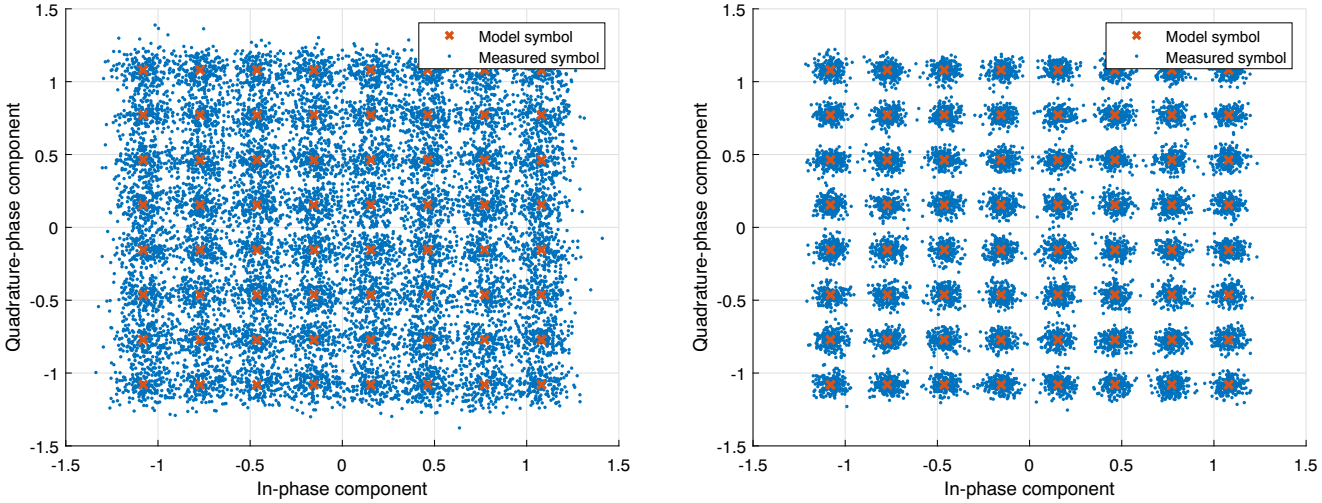
The linearization performance of the implemented algorithm with the SKY66293-21 and the HMC1114PM5E PA are illustrated in Fig. 5. The figures show the power spectral densities of the PA output signal without DPD and with the proposed DPD implementation, in addition to the absolute basic emission limits for the out-of-band power densities from Table 1. In order to fulfill these limits, the power densities of the transmit signals have to lay below the limits outside the BS RF bandwidth whatever the type of transmitter considered. The figures show that the absolute basic limits are satisfied with the algorithm implementation, with adequate difference between the limits and the out of band power densities of the predistorted signals, even on the edges of the signal band.

The EVM and ACPR of the measured signals without and with DPD are shown in Table 5. Since the effects of the nonlinear distortion are mitigated with the use of DPD, the EVM is consequently diminished in both PAs. Within the context of the 3GPP limits given in Table 1, neither of the PA outputs satisfy the EVM limit of 8 % without the DPD operation. However, with the use of DPD, the EVM in both cases falls well below the limit. Without the DPD, the only way to achieve the required 8 % limit would be to decrease the transmission power, therefore decreasing the power efficiency of the PA. With the DPD, the PAs can be operated more efficiently and could even be more saturated in terms of the EVM limit. EVM reduction is especially desirable when considering the HMC1114PM5E PA, since the initial condition without DPD is considerably over the limit. To further illustrate the effect the linearization has on the EVM,



(a) Skyworks SKY66293-21. (b) Analog Devices HMC1114PM5E.

Figure 5 Linearization performances obtained with the Skyworks' and Analog Devices' PAs, respectively. The specified 3GPP out-of-band absolute basic limits for local area (-32 dBm/MHz) and medium range (-25 dBm/MHz) base stations are also depicted in the figures.



(a) Without DPD (EVM = 11.31 %). (b) With proposed DPD (EVM = 6.16 %).

Figure 6 Phase and quadrature digital 64-QAM constellations of the Analog Devices' PA output signal when no DPD and the proposed DPD solution is applied.

Fig. 6 shows the in-phase and quadrature constellations of the HMC1114PM5E output without and with DPD. Although there are no requirements for the ACPR per se, the ACPR reduction is linked with the absolute basic limit set for the signals. Table 5 shows that the DPD algorithm reduces the ACPR with the SKY66293-21 to around -40 dB on both sides and with the HMC1114PM5E to around -37 dB, which in these test scenarios are enough to satisfy the absolute basic limits. Additionally, Fig. 7 presents the convergence of the $\boldsymbol{\alpha}_i$ coefficients as a function of the iteration number, in the measurements performed with the SKY66293-21 PA. It can be seen from the figure that all the coefficients are already converged in the last iteration of the DPD loop.

As is evident from Fig. 5a, the spectrum of the PA output is not symmetric, due to memory effects of the PA. The DPD implementation only has memory depth of $M = 1$, which might not be enough to compensate for such strong memory effects. Thus adding memory to the filters would very likely improve the performance of the algorithm. Another approach to improve the performance would be to use a more complex DPD model, such as the GMP [24]. Additionally, the whole algorithm utilizes 32-bit numeric precision, which also hinders the performance by introducing quantization noise to the calculations. Using wider bit widths would reduce the errors, thus potentially improving the performance. However, all these improvements would require additional hardware resources, which are not available on the current platform. Despite these shortcomings, the presented results demonstrate that the DPD implementation is capable of providing sufficient linearization of the PAs, and thus show its feasibility for 5G NR local and medium range base stations.

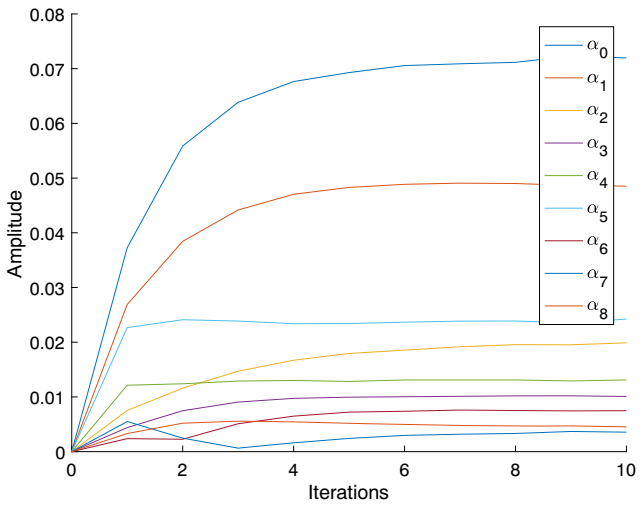


Figure 7 Coefficient $\boldsymbol{\alpha}_i$ convergence against iterations in the DPD learning stage.

6 Conclusions

In this paper, a high throughput DPD implementation and its functionality are presented. In order to achieve real-time operation, the predistortion of the transmit signal is executed within a GPU, while the adaptive part of the algorithm is implemented in a CPU core. The DPD implementation utilizes a dataflow computing framework, written in OpenCL for the GPU and C for the CPU. This configuration, combined with the low complexity of the DPD algorithm, allows for more than 400 Msamples/s sample rates for the DPD filtering, which is demonstrated on two different CPU/GPU combinations. The DPD implementation is shown to successfully linearize a 100 MHz 5G NR signal on two different power

amplifiers operating at 3.7 GHz, such that the in-band and out-of-band emission limits set by the latest 3GPP NR base station radio transmission and reception standards are satisfied.

Acknowledgements This work was supported by the Academy of Finland (under the projects #304147 "In-Band Full-Duplex Radio Technology: Realizing Next Generation Wireless Transmission", and #301820 "Competitive Funding to Strengthen University Research Profiles"), the Finnish Funding Agency for Innovation (Tekes, under the projects "5G Transceivers for Base Stations and Mobile Devices (5G TRx)" and "TAKE-5"), Nokia Networks, RF360 Europe, Pulse, Sasken, and Huawei Technologies, Finland.

References

1. (2018). 3GPP Tech. Spec. 38.104: NR; Base Station (BS) radio transmission and reception.
2. Abdelaziz, M. (2017). Reduced-complexity Digital Predistortion in Flexible Radio Spectrum Access. Ph.D. thesis, Tampere University of Technology.
3. Abdelaziz, M., Anttila, L., Kiayani, A., Valkama, M. (2017). Decorrelation-based concurrent digital predistortion with a single feedback path. *IEEE Transactions on Microwave Theory and Techniques*, *66*(1), 280–293.
4. Abdelaziz, M., Anttila, L., Tarver, C., Li, K., Cavallaro, J.R., Valkama, M. (2016). Low-complexity subband digital predistortion for spurious emission suppression in noncontiguous spectrum access. *IEEE Transactions on Microwave Theory and Techniques*, *64*(11), 3501–3517.
5. Afsardoost, S., Eriksson, T., Fager, C. (2012). Digital predistortion using a vector-switched model. *IEEE Transactions on Microwave Theory and Techniques*, *60*(4), 1166–1174.
6. Boutellier, J., & Hautala, I. (2016). Executing dynamic data rate actor networks on openCL platforms. In: 2016 IEEE International Workshop on Signal Processing Systems (siPS) (pp. 98–103). IEEE.
7. Cao, W., & Zhu, A. (2017). A modified decomposed vector rotation-based behavioral model with efficient hardware implementation for digital predistortion of RF power amplifiers. *IEEE Transactions on Microwave Theory and Techniques*, *65*(7), 2443–2452.
8. Ding, L., Zhou, G.T., Morgan, D.R., Ma, Z., Kenney, J.S., Kim, J., Giardina, C.R. (2004). A robust digital baseband predistorter constructed using memory polynomials. *IEEE Transactions on Communications*, *52*(1), 159–165.
9. Ghannouchi, F.M., & Hammi, O. (2009). Behavioral modeling and predistortion. *IEEE Microwave Magazine*, *10*(7), 52–64.
10. Guan, L., & Zhu, A. (2011). Simplified dynamic deviation reduction-based Volterra model for Doherty power amplifiers. In: 2011 Workshop on Integrated Nonlinear Microwave and Millimetre-wave Circuits (pp. 1–4). https://doi.org/10.1109/INMMIC.2011.5773325.
11. Haykin, S.S. (2005). Adaptive filter theory. Pearson Education India.
12. Intel Corp.: Intel Core i7-6700HQ Processor Product Specification. Tech. rep. (2015). https://ark.intel.com/content/www/us/en/ark/products/88967/intel-core-i7-6700hq-processor-6m-cache-up-to-3-50-ghz.html.
13. Intel Corp.: The Compute Architecture of Intel Processor Graphics Gen9, Version 1.0. Tech. rep. (2015). https://software.intel.com/sites/default/files/managed/c5/9a/The-Compute-Architecture-of-Intel-Processor-Graphics-Gen9-v1d0.pdf.
14. Jardin, P., & Baudoin, G. (2007). Filter lookup table method for power amplifier linearization. *IEEE Transactions on Vehicular Technology*, *56*(3), 1076–1087.
15. Kim, J., & Konstantinou, K. (2001). Digital predistortion of wideband signals based on power amplifier model with memory. *Electronics Letters*, *37*(23), 1.
16. Korpi, D., Choi, Y.S., Huusari, T., Anttila, L., Talwar, S., Valkama, M. (2015). Adaptive nonlinear digital self-interference cancellation for mobile inband full-duplex radio: Algorithms and RF measurements. In: 2015 IEEE Global Communications Conference (GLOBECOM) (pp. 1–7). IEEE.
17. Kwan, A.K., Helaoui, M., Boumaiza, S., Smith, M.R., Ghannouchi, F.M. (2009). Wireless communications transmitter performance enhancement using advanced signal processing algorithms running in a hybrid DSP/FPGA platform. *Journal of Signal Processing Systems*, *56*(2-3), 187–198.
18. Li, K., Ghazi, A., Tarver, C., Boutellier, J., Abdelaziz, M., Anttila, L., Juntti, M., Valkama, M., Cavallaro, J.R. (2017). Parallel digital predistortion design on mobile GPU and embedded multicore CPU for mobile transmitters. *Journal of Signal Processing Systems*, *89*(3), 417–430.
19. Logic Supply Inc.: Data sheet: Industrial Intel®Kaby Lake Fanless NUC Computer. Tech. Rep. ML100G-31 (2019). http://static.logicsupply.com/resources/spec-sheets/logic-supply-ml100-series-spec-sheet.pdf.
20. Luo, F.L. (2011). Digital front-end in wireless communications and broadcasting: circuits and signal processing. Cambridge University Press.
21. Ma, Y., Yamao, Y., Akaiwa, Y., Yu, C. (2013). Fpga implementation of adaptive digital predistorter with fast convergence rate and low complexity for multi-channel transmitters. *IEEE Transactions on Microwave Theory and Techniques*, *61*(11), 3961–3973. https://doi.org/10.1109/TMTT.2013.2281962.
22. Mateo, C., Carro, P.L., Garcia-Ducar, P., de Mingo, J., Salinas, I. (2017). Digital predistortion based on B-spline CPWL models in a RoF LTE mobile fronthaul. In: 2017 47Th European Microwave Conference (euMC), pp. 1136–1139. IEEE.
23. Mkadem, F., Islam, A., Boumaiza, S. (2016). Multi-band complexity-reduced generalized-memory-polynomial power-amplifier digital predistortion. *IEEE Transactions on Microwave Theory and Techniques*, *64*(6), 1763–1774.
24. Morgan, D.R., Ma, Z., Kim, J., Zierdt, M.G., Pastalan, J. (2006). A generalized memory polynomial model for digital predistortion of RF power amplifiers. *IEEE Transactions on Signal Processing*, *54*(10), 3852–3860.
25. Muhonen, K.J., Kavehrad, M., Krishnamoorthy, R. (2000). Look-up table techniques for adaptive digital predistortion: a development and comparison. *IEEE Transactions on Vehicular Technology*, *49*(5), 1995–2002.
26. Pascual Campo, P., Korpi, D., Anttila, L., Valkama, M. (2018). Nonlinear digital cancellation in full-duplex devices using spline-based hammerstein model. In: IEEE Global Communications Conference (GLOBECOM).

27. Paulo, S.D. et al. (2008). Adaptive filtering: algorithms and practical implementation. The International Series in Engineering and Computer Science, pp. 133–150.
28. Presti, C.D., Kimball, D.F., Asbeck, P.M. (2012). Closed-loop digital predistortion system with fast real-time adaptation applied to a handset WCDMA PA module. *IEEE Transactions on Microwave Theory and Techniques*, *60*(3), 604–618.
29. Stone, J.E., Gohara, D., Shi, G. (2010). OpenCL: A parallel programming standard for heterogeneous computing systems. *Computing in Science & Engineering*, *12*(3), 66.
30. Tehrani, A.S., Cao, H., Afsardoost, S., Eriksson, T., Isaksson, M., Fager, C. (2010). A comparative analysis of the complexity/accuracy tradeoff in power amplifier behavioral models. *IEEE Transactions on Microwave Theory and Techniques*, *58*(6), 1510–1520.
31. Wright, A.S., Klijsen, B.T., Yee, P.V., Hung, C.Y.K., Bennett, S.J. (2003). Digital predistortion methods for wideband amplifiers. US Patent 6,587,514.

Pablo Pascual Campo received his 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 pursuing his D.Sc. degree at Tampere University's Department of Electrical Engineering, Tampere, Finland. His research interests include digital predistortion, full-duplex systems and applications, and signal processing for wireless communications in the mmWave spectrum.

Vesa Lampu received his B.Sc. degree in Electrical Engineering in 2017 from Tampere University of Technology, Finland. He is currently working for the Department of Electrical Engineering of Tampere University, Finland as a research assistant and is finishing his M.Sc. thesis on a digital self-interference canceller algorithm implementation. His research interests include full-duplex and digital predistortion techniques and applications, real-time implementations and RF circuits.

Alexandre Meirhaeghe received the M.Sc. degree in electronics and computer engineering from INSA de Rennes, France, in 2018. In 2019, he joined as a consulting engineer ASTEK SA, a french company that supports their clients among diversified sectors with high value-added and bespoke results.

Jani Boutellier received the M.Sc. and Ph.D. degrees from the University of Oulu, Finland, in 2005 and 2009, respectively. Currently he is an Associate Professor at the School of Technology and Innovations, University of Vaasa, Finland. His research interests include dataflow programming, design and implementation of deep learning algorithms, and heterogeneous computing. He is a member of the IEEE Signal Processing Society DISPS Technical Committee.

Lauri Anttila received the M.Sc. and D.Sc. (Hons.) degrees in electrical engineering from Tampere University of Technology (TUT), Tampere, Finland, in 2004 and 2011. Currently, he is a University Researcher at the Department of Electrical Engineering at Tampere University (formerly TUT). His research interests are in 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. He has co-authored 100+ peer reviewed articles in these areas, as well as three book chapters.

Mikko Valkama was born in Pirkkala, Finland, on November 27, 1975. He received the M.Sc. and Ph.D. Degrees (both with honors) in electrical engineering (EE) from Tampere University of Technology (TUT), Finland, in 2000 and 2001, respectively. In 2003, he was working as a visiting post-doc researcher with the Communications Systems and Signal Processing Institute at SDSU, San Diego, CA. Currently, he is a Full Professor and Laboratory Head at the Laboratory of Electronics and Communications Engineering at TUT, Finland. His general research interests include radio communications, communications signal processing, estimation and detection techniques, signal processing algorithms for flexible radios, cognitive radio, full-duplex radio, radio localization, 5G mobile cellular radio networks, digital transmission techniques such as different variants of multicarrier modulation methods and OFDM, and radio resource management for ad-hoc and mobile networks.