

Research Article

Prototype Implementation of Two Efficient Low-Complexity Digital Predistortion Algorithms

Ernst Aschbacher,^{1,2} Mei Yen Cheong,³ Peter Brunmayr,² Markus Rupp,² and Timo I. Laakso^{3,4}

¹ MED-EL Medical Electronics, Research and Development, Fürstenweg 77a, 6020 Innsbruck, Austria

² Institute of Communications and Radio-Frequency Engineering, Vienna University of Technology, 1040 Vienna, Austria

³ Signal Processing Laboratory, Helsinki University of Technology, 02150 Espoo, Finland

⁴ National Board of Patents and Registration of Finland, 00101 Helsinki, Finland

Correspondence should be addressed to Ernst Aschbacher, ernst.aschbacher@medel.com

Received 1 February 2007; Revised 10 August 2007; Accepted 16 September 2007

Recommended by S. Gannot

Predistortion (PD) lineariser for microwave power amplifiers (PAs) is an important topic of research. With larger and larger bandwidth as it appears today in modern WiMax standards as well as in multichannel base stations for 3GPP standards, the relatively simple nonlinear effect of a PA becomes a complex memory-including function, severely distorting the output signal. In this contribution, two digital PD algorithms are investigated for the linearisation of microwave PAs in mobile communications. The first one is an efficient and low-complexity algorithm based on a memoryless model, called the simplicial canonical piecewise linear (SCPWL) function that describes the static nonlinear characteristic of the PA. The second algorithm is more general, approximating the pre-inverse filter of a nonlinear PA iteratively using a Volterra model. The first simpler algorithm is suitable for compensation of amplitude compression and amplitude-to-phase conversion, for example, in mobile units with relatively small bandwidths. The second algorithm can be used to linearise PAs operating with larger bandwidths, thus exhibiting memory effects, for example, in multichannel base stations. A measurement testbed which includes a transmitter-receiver chain with a microwave PA is built for testing and prototyping of the proposed PD algorithms. In the testing phase, the PD algorithms are implemented using MATLAB (floating-point representation) and tested in record-and-playback mode. The iterative PD algorithm is then implemented on a Field Programmable Gate Array (FPGA) using fixed-point representation. The FPGA implementation allows the pre-inverse filter to be tested in a real-time mode. Measurement results show excellent linearisation capabilities of both the proposed algorithms in terms of adjacent channel power suppression. It is also shown that the fixed-point FPGA implementation of the iterative algorithm performs as well as the floating-point implementation.

Copyright © 2008 Ernst Aschbacher et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Future mobile communication systems are intended to provide multimedia communications which require high-speed broadband transmissions. These systems have to make efficient use of the sparse and valuable spectrum while providing reliable communication. Linear signaling such as high-order quadrature amplitude modulation (QAM) is used as an efficient means to fulfill the high data rate requirement. Orthogonal frequency division multiplexing (OFDM) modulation is extensively employed and proposed for many broadband systems (e.g., WLAN, WiMax [1, 2], LTE of 3GPP [3]) due to its spectral efficiency and robustness in multipath envi-

ronments. The drawback of such schemes is their high peak-to-average power ratio (PAPR), which requires the transmitter system to be highly linear, especially the power amplifiers (PAs), in order to avoid nonlinear distortion. Nonlinear amplification produces in-band, as well as out-of-band distortion. While the increased error rate due to in-band distortion can be reduced using error correction coding, linearisation techniques are needed in order to limit the out-of-band power so that the stringent spectral mask requirements of such communications systems can be met.

With the use of a linearisation technique, nonlinear distortion can be compensated while the PA is driven into the nonlinear region to gain power efficiency. A remarkable

amount of research activities on linearisation techniques, both in analogue and digital domains, are notable in the literature of the past two decades. Examples of analogue linearisers are feedforward linearisation, Cartesian loop feedback lineariser [4] and PDs implemented using analogue components [5–7]. Digital linearisers are mainly predistortion based. In the late 1980s through the mid 1990s, many look-up table (LUT) based digital PDs were proposed [8–10]. LUT-based designs are limited by the slow adaptation due to their huge table size, especially when memory effects of the PA are considered.

Another type of digital PD is based on parametric models, in which the PD is described, for example, by a Volterra system [11], a polynomial function, a piecewise linear function or other PA model specific functions, such as the Saleh model [12]. The number of adaptive parameters is significantly reduced as compared to the LUT-based PD, so that the hardware complexity can also be kept low. Digital PD is advantageous compared to analogue schemes as it provides more flexibility (e.g., future system changes are more easily supported), and adaptivity is easy to incorporate. It is also more robust, for instance, its linearisation performance does not depend on difficult to tune analogue components as in the feedforward linearisation method [4]. Digital PDs also offer higher linearity, as well as better power efficiency and cost effectiveness compared to their analogue counterparts. Recently, digital baseband PDs have become more feasible than before due to the rapid improvement of digital signal processing (DSP) technology.

Most of the PDs proposed in the literature are validated by computer simulations and the PA to be linearised is often an analytical or characteristic nonlinear function. However, implementation of the PD algorithm on hardware and evaluation based on measurement of the actual linearisation of a practical PA better describes the behavior of a proposed PD. There are only a handful of publications which considered hardware implementation and validation of the PDs based on measurement of practical PAs. For example, [13–16] reported implementation of LUT-based digital PDs on DSP/FPGA hardware and validated on real PAs in measurement testbeds. Another example of a partial hardware implementation of a parametric model PD is reported in [17], where the training algorithm of a memory polynomial PD is implemented on a Texas Instruments' floating-point digital signal processor (TMS320C67xx). In [18] crest-factor reduction and digital predistortion are evaluated in a record-and-playback fashion, but not using a fixed-point and real-time hardware implementation. Also in [19] a memory polynomial PD is evaluated on a PA in a record-and-playback mode.

In this paper, two parametric models, which are rather different in their nature, are considered for modeling the digital PDs. One is the simplicial canonical piecewise linear (SCPWL) function, which is suitable for modeling memoryless nonlinearities. The linear affine property of the SCPWL function is exploited for developing a computationally efficient PD identification algorithm. The SCPWL PD parameters are identified without involving complex numerical computation such as matrix inversion. Another is the Volterra series that is suitable for modeling nonlinearities

with memory. As the pre-inverse of the Volterra model PA is difficult to obtain analytically, iterative methods based on the Newton-Raphson method and successive approximation method are employed to identify the Volterra model PD. The secant method instead of the standard Newton-Raphson method is used in order to relax the requirement for an analytic PA model and to reduce the computational burden on computing the step size. Convergence analysis by simulations for these iterative methods is provided.

A measurement testbed was built for measuring, testing, and prototyping of the PD algorithms. The nonlinear characteristics of a test PA (Minicircuits MC-ZVE8G [20]) was measured. The input-output data obtained by exciting the test PA with a broadband multitone signal is used for identification of the PDs. Then the performance of the identified PDs in linearising the test PA is evaluated by measurement. The testbed also provides facilities for the chosen PD algorithm to be implemented on digital hardware. An iterative PD algorithm was implemented on an FPGA. Measurement results prove excellent linearisation quality.

This paper is organized as follows. In Section 2, we motivate the need for PD linearisers in communications systems and formulate the PD problem. Section 3 gives an overview of the nonlinear models with and without memory considered for modeling the PA and PD in this paper. The proposed PD algorithms are presented in Section 4 followed by the setup of the measurement testbed in Section 5. In Section 6, the linearisation performance of the PDs is evaluated in the offline measurement mode. Section 7 discusses the FPGA implementation of the iterative Volterra model PD. Measurement results of the PD running in real-time on an FPGA are presented in this section as well. Conclusions are drawn in Section 8.

Notation

Discrete-time signal sequences are denoted by italic small cap font with the time index denoted by n within square brackets, for example, $x[n]$. Signal operators are denoted by upper-case blackboard font, for example, $\mathbb{H}\{\cdot\}$ in $y[n] = \mathbb{H}\{x[n]\}$. The operator \mathbb{H} (generally a nonlinear operator in this paper) transforms the signal $x[n]$ into the signal $y[n]$. Scalar functions are denoted by italic small cap font with argument within parentheses, for example, $f(\cdot)$. Vectors are in lower-case boldface letters and matrices are in upper-case boldface letters. Signals are in general complex-valued unless otherwise stated.

2. MOTIVATION AND PROBLEM FORMULATION

Power efficiency and linearity of the power amplifier (PA) are two equally important but contradicting requirements in mobile communications systems. If the PA system in the base station is operated inefficiently, the maintenance costs and power consumption will become significantly higher and the life span of the PA will also be reduced. Power efficiency is particularly important in the mobile units for prolonging the battery life. However, due to intrinsic properties, power efficient PAs are nonlinear. Nonlinear distortion results in

in-band signal distortion and spectral regrowth in the amplified signal. These effects lead to increased bit-error rate at the receiver and violation of regulatory specifications on adjacent channel power (see, e.g., [21]).

The efficiency of a radio-frequency (RF) PA is usually measured by the *power-added efficiency* (PAE)

$$\eta = \frac{P_{\text{RF,out}} - P_{\text{RF,in}}}{P_{\text{DC}}}, \quad (1)$$

whereby $P_{\text{RF,out}}$ and $P_{\text{RF,in}}$ denote the RF output and RF input powers of the PA, respectively, and P_{DC} is the supplied DC power. It measures how efficient DC power is converted to RF output power, excluding the power due to the RF input signal. In a system that transmits signals with fluctuating envelope, for example, OFDM or CDMA signals, a significant amount of power back-off (reducing $P_{\text{RF,in}}$) is typically required in order to limit nonlinear distortion caused by the PA. However, when power back-off is imposed, power efficiency is reduced. This can be observed from the simple relationship in (1). When the input signal power is reduced, the effective RF output power, that is, the numerator in (1), decreases while P_{DC} remains constant, leading to a reduced PAE. The typical values of PAE achieved in today's PAs for 3G mobile communication base stations without linearisation (operated in the linear region) are around 20%, whereas PAs in handsets achieve around 40% efficiency [22]. Therefore, in order to meet regulatory requirements on adjacent channel power and signal quality while operating the PA power efficiently, linearisation techniques are required. In this paper digital predistortion linearisers are considered.

2.1. Formulation of the predistortion problem

In designing the PD, the relationship between the nonlinear system and the PD has to be established first. Figure 1 illustrates the discrete-time, baseband equivalent system of a predistortion filter \mathbb{P} placed in cascade with a nonlinear system \mathbb{N} . The lower branch represents an ideal linear PA \mathbb{L} where the output is $d[n] = \mathbb{L}\{u[n]\} = g \cdot u[n - \Delta]$. The nonlinear system \mathbb{N} may include the digital-to-analogue converter, I-Q modulators, RF mixer, and most importantly the PA system which may be of single or multiple stages. The predistortion filter \mathbb{P} should be designed such that the output $y[n]$ is as close as possible to the linearly amplified (and delayed) version of the input signal, that is,

$$y[n] = \mathbb{N}\{\mathbb{P}\{u[n]\}\} \approx d[n] = \mathbb{L}\{u[n]\} = g \cdot u[n - \Delta]. \quad (2)$$

Here, Δ denotes the introduced delay and g is the targeted linear gain. Note that \mathbb{P} is the pre-inverse filter of \mathbb{N} . In order to identify the predistortion filter \mathbb{P} , the nonlinear system \mathbb{N} is first modeled and expressed as a nonlinear function. In this paper two nonlinear functions, that is, the simplicial canonical piecewise linear function and the Volterra series are employed for modeling \mathbb{N} . Then algorithms are devised to find the pre-inverse \mathbb{P} of these functions, that is, the PDs. The PD identification algorithms are presented in Section 4.

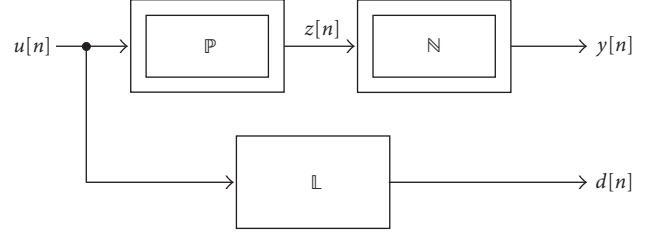


FIGURE 1: Linearisation problem.

Next, a simplified description of how a digital PD is put in operation in practice is given. Figure 2 shows a block diagram of a typical transmitter employing a digital predistortion (DPD) system. The input signal $u[n]$, consisting of the in-phase $I[n]$ and quadrature-phase component $Q[n]$ is pre-filtered by a nonlinear predistortion filter. After digital-to-analogue conversion the signals modulate the carrier at the transmit frequency f_c . Before transmission, this analogue RF signal is amplified by a power amplifier. Ideally, a feedback path is used to feed the output signal back to the PD identification algorithm in order to track the behaviour fluctuation of the PA due to temperature variation, aging, or changing of operational mode, for example, in multichannel PAs. Then, the transmitted signal is a linearly amplified version of the input signal if the PD is properly identified.

3. POWER AMPLIFIER MODELS

This section presents the two functions used in this work for modeling the PA and subsequently the PD. First, the simplicial canonical piecewise linear function (SCPWL) which is suitable for modeling static nonlinearities is presented. Following, the Volterra series, which can be used to model nonlinearities with memory, is presented.

3.1. Static model: SCPWL function

A piecewise linear (PWL) function is a function that divides the input space into a finite number of partitions, each described by a linear affine function. Conventional PWL functions are expressed region by region and thus require a huge amount of coefficients. A compact form known as the canonical PWL function was first introduced in [23]. It is expressed as a global function with much fewer coefficients than the conventional PWL function. More recently, the concept of simplicial partition is used in [24] to develop PWL functions in an even more compact form. This class of PWL functions is known as the *simplicial canonical piecewise linear* (SCPWL) functions. PWL functions have been used for modeling and analysis of nonlinear circuits [25, 26] but are still uncommon for modeling PA nonlinearities.

There are a few advantages of modeling static nonlinearities using a PWL function compared to a polynomial. With proper partitioning of the input space, the PWL function can approximate strong nonlinearities (sharp compression/expansion) more accurately. It does not pose numerical problems such as the Runge phenomenon [27] exhibited

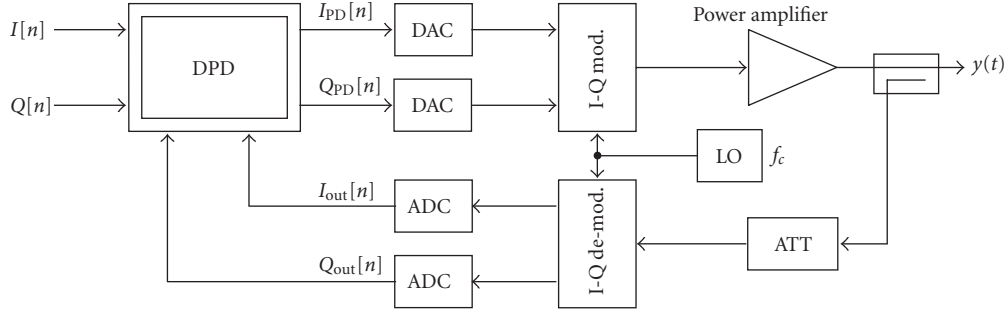


FIGURE 2: Concept of digital predistortion.

by high-order polynomials. Moreover, parameter estimation for polynomials often involves inversion of a Vandermonde matrix which is usually ill-conditioned. In the contrary, the structure provided by the linear affine property of a PWL function allows an efficient parameter estimation algorithm which does not involve matrix inversion [28].

The SCPWL function [24] in \mathbb{R}^1 with positive real input r is expressed as

$$f_{\beta}(r) = c_0 + \sum_{i=1}^{\sigma-1} c_i \lambda_i(r) = \mathbf{c}^T \Lambda_{\beta}(r), \quad (3)$$

where $\Lambda_{\beta}(r) = [1, \lambda_1(r), \dots, \lambda_{\sigma-1}(r)]^T$ is the basis function vector and $\mathbf{c} = [c_0, \dots, c_{\sigma-1}]^T$ is the SCPWL coefficient vector. The breakpoints $\beta = [\beta_1, \beta_2, \dots, \beta_{\sigma}]^T$ are predefined and can be chosen to optimally fit a given nonlinear function, σ is the number of breakpoints. In (3), the subscript in $\Lambda_{\beta}(r)$ and $f_{\beta}(r)$ indicates the chosen set of breakpoints for a given nonlinearity that the SCPWL function is modeling. The i th basis function is given as

$$\lambda_i(r) = \begin{cases} \frac{1}{2}(r - \beta_i + |r - \beta_i|), & r \leq \beta_{\sigma}, \\ \frac{1}{2}(\beta_{\sigma} - \beta_i + |\beta_{\sigma} - \beta_i|), & r > \beta_{\sigma}. \end{cases} \quad (4)$$

The SCPWL function is suitable for modeling static nonlinearities such as AM/AM and AM/PM functions. Let the baseband input and output signals be represented by $z[n] = r_z[n]e^{j\varphi_z[n]}$ and $y[n] = r_y[n]e^{j(\varphi_z[n] + \varphi[n])}$, where $r_z[n]$ and $r_y[n]$ denote the magnitude of the input and output signals, respectively. Then the AM/AM and AM/PM conversions can be approximated using two SCPWL functions as

$$\begin{aligned} f_r(r_z[n]) &= r_y[n] = \mathbf{c}_r^T \Lambda_{\beta_r}(r_z[n]), \\ f_{\varphi}(r_z[n]) &= \varphi[n] = \mathbf{c}_{\varphi}^T \Lambda_{\beta_{\varphi}}(r_z[n]), \end{aligned} \quad (5)$$

where β_r and β_{φ} are the breakpoints vectors of the AM/AM and AM/PM functions, respectively.

3.2. Dynamic model: Volterra series

The Volterra series is known as the most complete function for describing dynamic nonlinear systems [29, 30]. It is a

functional power series of the form (if not specified, integration and summation limits are from $-\infty$ to ∞)

$$\begin{aligned} y(t) &= \mathbb{H}\{z(t)\} \\ &= h_0 + \sum_{p=1}^{\infty} \int \cdots \int h_p(t, \tau_1, \dots, \tau_p) \\ &\quad \times z(\tau_1) \cdots z(\tau_p) d\tau_1 \cdots d\tau_p, \end{aligned} \quad (6)$$

in which \mathbb{H} is a nonlinear functional of the continuous function $z(t)$, h_0 is a constant, t is a parameter, and $h_p(\cdots)$, $p \geq 1$, are continuous functions, called the Volterra kernels. If $p = 1$ the Volterra series reduces to the input-output representation of a simpler system:

$$y(t) = h_0 + \int h_1(t, \tau_1) z(\tau_1) d\tau_1. \quad (7)$$

If furthermore $h_0 = 0$, a linear system is obtained and the Volterra series reduces to a convolution. A Volterra series describes a large class of nonlinear systems, namely, all continuous nonlinear systems with fading memory [31]. Here, a truncated and stationary Volterra series is used to model the power amplifier. Taking into account the bandpass nature of the power amplifier, the discrete-time complex baseband Volterra model of the power amplifier is [32]

$$\begin{aligned} y[n] &= \mathbb{N}\{z[n]\} \\ &= \sum_{p=0}^{P-1} \mathbb{H}_{2p+1}\{z[n]\} = \sum_{p=0}^{P-1} \sum_{\mathbf{n}_{2p+1} \in \mathbb{N}^{2p+1}} h_{2p+1}[\mathbf{n}_{2p+1}] \\ &\quad \times \prod_{i=1}^{p+1} z[n - n_i] \prod_{i=p+2}^{2p+1} z^*[n - n_i]. \end{aligned} \quad (8)$$

For notational compactness, the vector $\mathbf{n}_{2p+1} = [n_1, \dots, n_{2p+1}]^T$ is used. This model can be easily simplified to the static case (i.e., memoryless), where the kernels reduce to scalars:

$$y[n] = e^{j \arg\{z[n]\}} \sum_{p=0}^{P-1} h_{2p+1}|z[n]|^{2p+1} = e^{j \arg\{z[n]\}} f(r_z[n]). \quad (9)$$

The (complex) nonlinear transformation can be rewritten as

$$f(r_z[n]) = f_r(r_z[n])e^{jf_\phi(r_z[n])}, \quad (10)$$

with the AM/AM transformation $f_r(r_z[n]) = |f(r_z[n])|$ and the AM/PM conversion $f_\phi(r_z[n]) = \arg\{f(r_z[n])\}$. The P complex parameters h_{2p+1} , $p = 0, \dots, P-1$, are the model parameters and describe the AM/AM, as well as the AM/PM conversion.

4. PREDISTORTION FILTERS

This section discusses the PD identification algorithms. A non-iterative method known as the image coordinate mapping (ICM) method [28] is employed for identifying the SCPWL PD. The ICM method is discussed in Section 4.1. Two iterative methods are considered for approximating the pre-inverse of the Volterra model PD, one based on the Newton-Raphson method and the other is a successive approximation method. The iterative methods are presented in Section 4.2 together with the analysis of their convergence behaviour.

4.1. Identification of the SCPWL PD: non-iterative solution

The ICM method is developed by exploiting the linear affine property of the SCPWL function. The ICM method is founded on the mirror image resemblance of the PA and PD's static nonlinearities along the unit linear gain line. When the static nonlinearity of a PA is modeled using a PWL function, each linear affine subregion is defined by a straight line connecting two coordinates. Based on this property, the PWL subregions of the PD can be obtained by finding the mirror images of the coordinates that define these linear affine functions of the PA. The concept of vector projection (in this case, reflection) using a transformation matrix is used in the ICM method [28] for finding the PD coordinates.

Consider a unit desired linear gain at the output of the PD-PA cascade. The transformation of \mathbf{b} to the image coordinates \mathbf{b}' as shown in Figure 3(a) can be performed using a 2-by-2 antidiagonal matrix with the nonzero elements equal one as

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \quad (11)$$

This transformation swaps the input and output of the PA. In effect, the mirror image connotes an inverse function of the PA. However, in practice, the desired linear gain is rarely chosen as one.¹ For non-unity linear gain, the PD function is not an exact mirror image of the PA. The input-output relation of the PD's linear affine functions must also take into account the desired linear gain g . This amplification factor

can be incorporated either by multiplying the output of the PD by g or dividing the input of the PD by g . Notice that the output space of the PD must coincide with the input space of the PA. The gain must therefore be incorporated in the input range of the PD. Thus, the ICM matrix for an arbitrary desired linear gain g is given as

$$\mathbf{Q} = \begin{bmatrix} 0 & \frac{1}{g} \\ 1 & 0 \end{bmatrix}. \quad (12)$$

The PD coordinates are then obtained as

$$\mathbf{b}' = \mathbf{Q}\mathbf{b}. \quad (13)$$

Figure 3(b) shows an example of the nonlinear characteristic of the SCPWL PD with respect to the PA characteristic when $g = 1.2$.

Once all the image coordinates \mathbf{b}'_k (for $k = 1, \dots, \sigma$) are obtained, the breakpoints for the PD β' and the corresponding amplitude responses $f_{\beta'}(\mathbf{r} = \beta')$ are obtained. Substituting into (3), the SCPWL function for the PD can now be written as

$$f_{\beta'}(\mathbf{r}_i = \beta'_i) = \Lambda_{\beta'}^T(\mathbf{r}_i = \beta'_i)\mathbf{c}', \quad (14)$$

where \mathbf{c}' is the coefficients vector of the PD that needs to be identified. By collecting (14) for $i = 1, \dots, \sigma$ into matrix-vector form, we have

$$\mathbf{f}_{\beta'}(\mathbf{r} = \beta') = \mathbf{L}_{\beta'}(\mathbf{r} = \beta')\mathbf{c}', \quad (15)$$

where the matrix $\mathbf{L}_{\beta'}(\beta') = [\Lambda_{\beta'}(\beta'_1), \Lambda_{\beta'}(\beta'_2), \dots, \Lambda_{\beta'}(\beta'_\sigma)]^T$ is the basis function matrix evaluated at the PD partition points β' .

Note that $\mathbf{L}_{\beta'}(\beta')$ is a nonsingular square matrix. The inverse can be obtained by performing some linear operations on $\mathbf{L}_{\beta'}(\beta')$. It is shown in [33] that its inverse $\mathbf{L}_I(\beta') \equiv \mathbf{L}_{\beta'}^{-1}(\beta')$ has nonzero elements only on the main diagonal and two lower diagonals. Due to the linear affine property of the SCPWL function, these nonzero elements can be computed from the knowledge of the partition points β' . This computation involves only subtractions and divisions. Thus, the SCPWL PD coefficients can be obtained without invoking matrix inversion as

$$\mathbf{c}' = \mathbf{L}_I(\beta')\mathbf{f}_{\beta'}(\beta'), \quad (16)$$

with low computational complexity.

4.2. Identification of the Volterra PD: iterative solution

As mentioned earlier, PD models are identified as the pre-inverse of the PA model. In general, the pre-inverse systems of nonlinear systems with memory, for example, the Volterra model considered in this paper, are not easily determined analytically. In [34] a method for the construction of the p th-order pre-inverse filter for Volterra systems is introduced. However, this method is rather complicated, which makes it unsuitable for practical implementation. Instead of identifying the model parameters of the PD, iterative methods can be used to find the predistorted signals directly.

¹ A reasonable choice of the desired linear gain is to choose a value that leads to a maximum linearisation range, for example, up to the saturation point of an AM/AM characteristic.

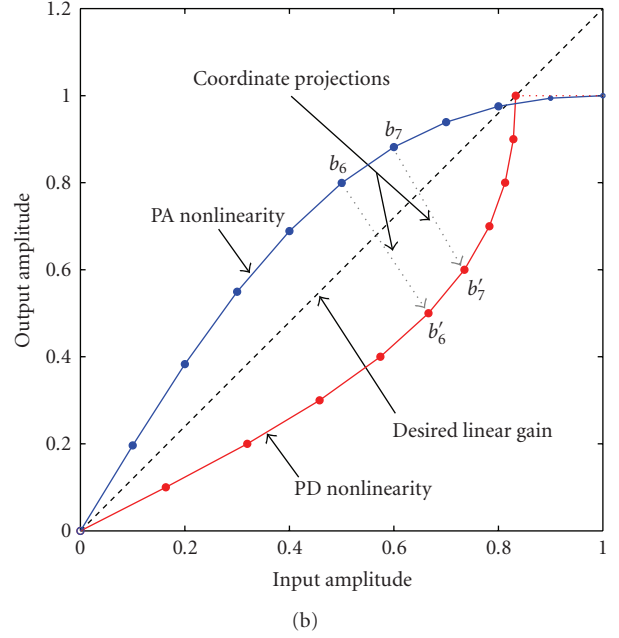
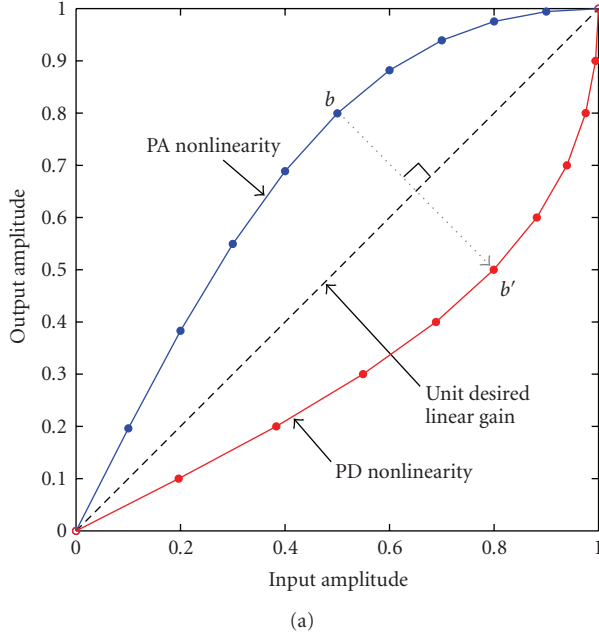


FIGURE 3: Mirror image resemblance of PA and PD nonlinearities.

4.2.1. Root search: secant method

By reorganizing the relationship of the nonlinear system and the PD in (2) to

$$\mathbb{N}\{z[n]\} - g \cdot u[n - \Delta] = \mathbb{T}_u\{z[n]\} = 0, \quad (17)$$

the problem of finding the predistortion filter \mathbb{P} is reformulated. The task is now to search the root $z_*[n]$ of (17), which is the output of the predistortion filter, see Figure 1. For most nonlinear operators \mathbb{N} (here, \mathbb{N} is the power amplifier model), an analytic solution is not known. But the root $z_*[n]$ can be searched iteratively which gives an approximate solution. A common method to solve nonlinear equations, which can also be applied to functionals, is the Newton-Raphson method [35]. In this case the iterative algorithm reads

$$z_{i+1}[n] = z_i[n] - \frac{1}{\partial_z \mathbb{N}\{z_i[n]\}} \mathbb{T}_u\{z_i[n]\}, \quad i \geq 0. \quad (18)$$

The advantage of the Newton-Raphson method is its rapid convergence. In the neighbourhood of the solution, the method converges with quadratic order. If $\varepsilon_i[n] = \|z_i[n] - z_*[n]\| / \|z_*[n]\|$ denotes the relative error at iteration-step i , then

$$\varepsilon_{i+1}[n] \sim \varepsilon_i[n]^2. \quad (19)$$

This rapid convergence is achieved at a high computational cost since the reciprocal value of $\partial_z \mathbb{N}\{z_i[n]\}$ has to be computed. Convergence of the Newton-Raphson method cannot be guaranteed but is generally achieved if the initial guess $z_0[n]$ is not too far from the solution $z_*[n]$.

Furthermore, notice that this method requires the derivative of the PA model $\partial_z \mathbb{N}$ to be evaluated at $z_i[n]$, that

is, the model has to be analytic. Most PA models, for example, (8), are not analytic (see, e.g., the special case for the static model (9)—the function $|z[n]|$ is analytic only at $z[n] = 0$). Since the Newton-Raphson method is not applicable to the Volterra PA model, an alternative algorithm is searched for. The Newton-Raphson step size can be approximated using the secant method. In this case $\mathbb{T}_u\{z[n]\}$ need not be analytic. The iterative secant algorithm reads

$$z_{i+1}[n] = z_i[n] - \frac{z_i[n] - z_{i-1}[n]}{\mathbb{N}\{z_i[n]\} - \mathbb{N}\{z_{i-1}[n]\}} \mathbb{T}_u\{z_i[n]\}, \quad (20)$$

$i \geq 0, z_{-1}[n], z_0[n] \text{ given.}$

The derivative $\partial_z \mathbb{N}\{z_i[n]\}$ is approximated with the secant. The complexity is significantly reduced compared to the standard Newton-Raphson method, since for the calculation of the secant, only $\mathbb{N}\{z_i[n]\}$ has to be calculated. But this has to be computed in any case for the calculation of $\mathbb{T}_u\{z_i[n]\}$ (cf. (17)).

Two initial values are needed. Since it is expected that the solution is only slightly different from the input signal (as long as the power amplifier is not heavily nonlinear), the input signal $z_0[n] = u[n]$ is used. The second initial value $z_{-1}[n] = 0$, for simplicity. Also this algorithm is not guaranteed to converge. The convergence depends on the initial values $z_{-1}[n]$ and $z_0[n]$ —if they are sufficiently close to the solution the algorithm converges. It is shown, for example, in [36], that the convergence rate is

$$\varepsilon_{i+1}[n] \sim \varepsilon_i[n]^\phi, \quad (21)$$

whereby $\phi = (1/2)(1 + \sqrt{5}) \approx 1.618$ is the golden ratio. It is slower than the convergence rate of the Newton-Raphson

method but can be improved if instead of $z_{i-1}[n]$ in (20) a value closer to $z_i[n]$ is used, for example,

$$\tilde{z}_{i-1}[n] = \lambda z_i[n] + (1 - \lambda)z_{i-1}[n], \quad \lambda \in [0, 1). \quad (22)$$

As λ approaches one, the derivative is better approximated with the secant. For simplicity of the hardware realization, the conventional secant algorithm with $\lambda = 0$ is used in both the offline MATLAB and the real-time FPGA implementations (see Sections 5–7).

4.2.2. Fixed-point search: successive approximation

The problem of determining the PD filter can be reformulated in yet another way [37]. If the nonlinear model \mathbb{N} allows for an additive decomposition, that is,

$$\mathbb{N}\{z[n]\} = \mathbb{H}_1\{z[n]\} + \sum_{p=1}^{P-1} \mathbb{H}_{2p+1}\{z[n]\}, \quad (23)$$

the problem (2) can be rewritten as a fixed-point equation in $z[n]$ as

$$z[n] = \mathbb{H}_1^{-1} \left\{ g \cdot u[n - \Delta] - \sum_{p=1}^{P-1} \mathbb{H}_{2p+1}\{z[n]\} \right\} = \mathbb{S}_u\{z[n]\}. \quad (24)$$

The fixed-point $z[n]$ is the output of the PD filter for the input $u[n]$. This fixed-point is determined iteratively with the method of successive approximation [35, 37]

$$z_{i+1}[n] = \mathbb{S}_u\{z_i[n]\}, \quad i \geq 0, \quad z_0[n] \text{ is given.} \quad (25)$$

This method can only be used if the problem can be brought into a fixed-point equation in terms of $z[n]$. This is possible for models that allow for an additive decomposition like (23) and where the first term \mathbb{H}_1 can be inverted, for example, Volterra models with a linear part that can be inverted. Other nonlinear models may not allow such a fixed-point formulation.

The advantage of the successive approximation method compared with the secant method is that the convergence analysis can be performed using the contraction mapping theorem [37]. It provides a sufficient condition for convergence and states that the successive approximation converges to the fixed-point if the operator \mathbb{S}_u is contractive on a closed set of a Banach space [35]. This convergence analysis is technically complex, for instance, the norms of the operators \mathbb{H}_{2p+1} in (24) have to be determined in order to ascertain that the operator \mathbb{S}_u is contractive. In practice the norms can only be upper-bounded, so that the analysis gives in general rather conservative results which are often not very helpful in practice.

The convergence rate of successive approximation is linear, that is,

$$\varepsilon_{i+1}[n] \sim \varepsilon_i[n], \quad (26)$$

thus is much smaller than the convergence rate of the Newton-Raphson or secant method. The consequence is

that more iterations have to be performed for achieving a certain linearisation accuracy compared to the former two methods, meaning that hardware complexity is increased. In Section 4.2.3 it is shown by simulations that for a certain linearisation accuracy more iterations have to be performed with successive approximation compared to the secant method.

4.2.3. Convergence rate

In order to compare the convergence rate of the two methods, the secant method and the successive approximation, an example Volterra model is linearised. The parameters of the Volterra model are obtained using input/output data generated with an RF-circuit simulation using ADS [38]. The simulated PA is a Motorola LDMOS amplifier (MRF21125). Based on this data (WCDMA input signal, one channel) the parameters of a Volterra model \mathbb{N} (cf. (8)) are estimated. This assures that the example system to be linearised is realistic. The Volterra model is of fifth-order and each kernel has a memory length of two samples (sampling rate is $3.84 \text{ MHz} \times 8 = 30.72 \text{ MHz}$). In total 20 (complex) parameters are necessary. The linearisation error is defined as

$$J_{\text{lin}}(i)[\text{dB}] = 10 \log \left(\frac{\|e_{\text{lin},i}[n]\|_2^2}{\|d[n]\|_2^2} \right), \quad (27)$$

with

$$e_{\text{lin},i}[n] = y_i[n] - d[n] = \mathbb{N}\{z_i[n]\} - g \cdot u[n - \Delta], \quad (28)$$

whereby $z_i[n]$ is calculated with the secant method (20) or with successive approximation (25) and applied to the PA model $\mathbb{N}\{\cdot\}$. According to (21) the error decreases with every iteration step by approximately 16 dB if the secant method is used, whereas with successive approximation the error decreases with approximately 10 dB per iteration, corresponding to the linear convergence behaviour of this method, see (26). Figure 4 presents a graphical illustration.

Due to the slow convergence, the successive approximation method is too costly in terms of hardware resources for implementation in an FPGA. Therefore, only the secant method is implemented. The successive approximation method is presented here for comparison.

5. THE PROTOTYPING SYSTEM

In this work, the proposed PDs are designed using measurement data obtained by exciting the Minicircuits MC-ZVE8G [20] test PA with a broadband multisine signal. Then performance of the PD algorithms on linearising the test PA is evaluated by measurements. In this section, the setup of the measurement testbed is first presented. Then, the two test modes for testing the PD algorithms, namely, the offline test and real-time test, are defined. The limitations of the measurement testbed are also briefly discussed.

5.1. Measurement testbed

The testbed used in the work for measurements, testing, and prototyping consists of a digital signal processing (DSP) part

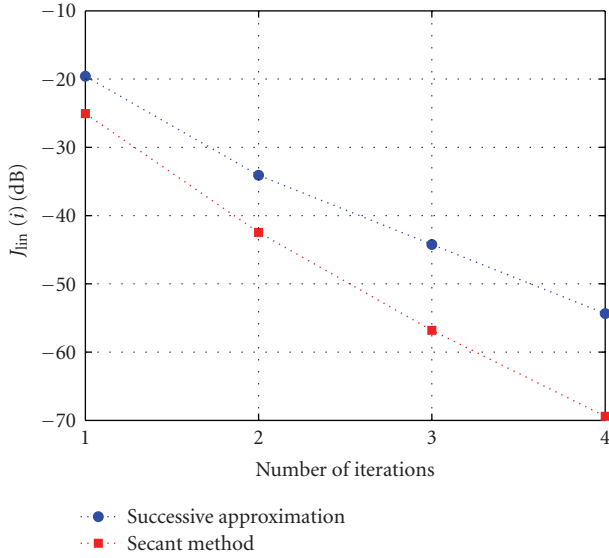


FIGURE 4: Comparison of the convergence rate of the secant method and the method of successive approximation.

and a radio frequency (RF) processing part. The DSP part is built up with a host computer and DSP hardware, and the RF part includes basic RF transceiver hardware and the test PA MC-ZVE8G. In the following, the setup of these two parts is detailed.

5.1.1. Digital signal processing part

Figure 5 illustrates the DSP part with hardware involved in the testbed. The interface between the host computer and the DSP hardware is provided by the Sundance SMT310Q [39] peripheral component interface (PCI) card that carries all DSP hardware on it.

Two Sundance SMT351-G memory modules [40] are mounted on this carrier board, giving a total of 2 GB memory for input-output (IO) data storage. The Sundance SMT370-AC [41] module provides the ADC/DAC functions. This module is equipped with the AD9777 [42] DAC from Analog Devices which implements also a digital I-Q modulator. Using this I-Q modulator, the baseband signal is digitally modulated onto an intermediate frequency (IF) carrier (center frequency 70 MHz) before DA conversion. The Sundance SMT370-AC module is also equipped with a Xilinx Virtex-2 XC2V1000 FPGA [43], which allows a proposed PD algorithm to be implemented and tested in real time.

The Sundance SMT365 digital signal processor (DSP) module configures all other modules. It configures the ADC/DAC and commands data transfer from the host computer to the memory module and then to the SMT370-AC module and vice versa. When the PD algorithm is implemented on the FPGA, it sets the model parameters of the PD filter on the FPGA after each update of the parameters set.

5.1.2. Radio frequency part

The block diagram of the RF part of the testbed is shown in Figure 6. In the transmit path, an attenuator is placed before the up-converter to reduce the power of the transmitted signal. This is done to minimize the nonlinear effect caused by the up-converter. Then the signal is mixed to a center frequency $f_c = 2.45$ GHz and filtered. A preamplifier is used to amplify the signal at the output of the up-converter to a sufficient level. An adjustable attenuator is used to control the input-power backoff (IBO) level of the signal to the test PA.

After the PA, the signal is fed back to the receive path. Again, the output signal of the PA is attenuated to ensure linearity of the down-converter. A common local oscillator is used for both the up-converter and the down-converter in order to avoid phase imbalance. The signal is down-converted to IF and filtered. The IF signal is amplified before the ADC so that the dynamic range of the ADC is optimally utilized.

5.2. Test modes

In this work, the proposed PDs in Section 4 are first identified and tested using a synthetic PA model in MATLAB. The linearisation performance is measured by the adjacent channel power ratio (ACPR) of the PA output signal. In the simulated environment, the power spectral density of the PA output signal showed that the proposed PD algorithms to be evaluated on a practical PA were successful in suppressing the ACPR.

Next, the PD algorithms are brought to test on a practical PA MC-ZVE8G on the testbed. A spectrum analyzer is used to examine the linearisation performance based on the ACPR of the PA output signal. The testbed supports two test modes for testing the performance of the proposed PDs, namely, the offline mode and the real-time mode. The configuration of the RF part is common for the two test modes. In both test modes, the nonlinear characteristics of the PA (modeled using an SCPWL function or a Volterra filter) are identified in the host computer using algorithms implemented in MATLAB. Different configurations in the DSP part that determine the test mode are as follows.

In the offline mode, the PDs are also identified in the host computer. Then, the input data is predistorted with the identified PD and transferred back to the memory module. In this mode, the predistorted signal is computed using double-precision floating-point arithmetic in MATLAB. From the memory, the predistorted signal is transmitted directly to the DAC and subsequently to the PA via the RF part. The FPGA is bypassed. The offline test examines the PD performance in a record-and-playback fashion. Both the SCPWL PD and the Secant-Volterra PD are tested in this mode. The results of the offline test are discussed in Section 6.

In the real-time mode, the PD algorithm is implemented on the FPGA. The PA model parameters identified in the host computer are transferred to the FPGA for implementation of the PD filter. Then, the excitation signal data is sent to the memory without being predistorted. From the memory, the data is transmitted through the PD filter on the FPGA and

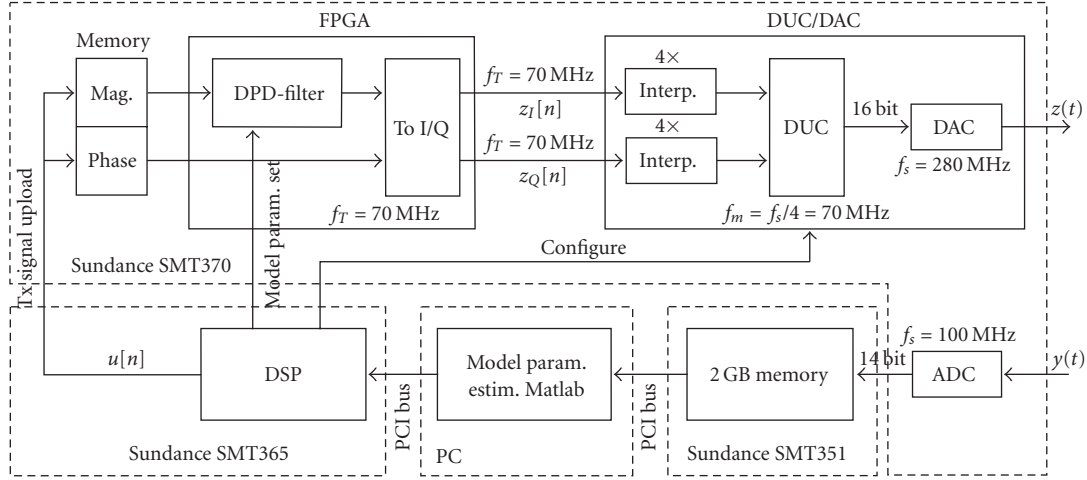


FIGURE 5: DSP part of the testbed.

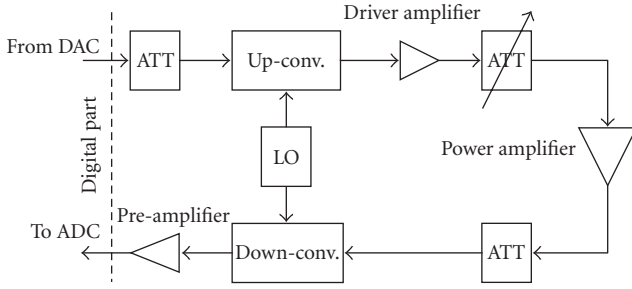


FIGURE 6: RF part of the testbed.

predistorted in a real-time manner, see Figure 5. Then the data is sent to the PA to examine the linearisation performance. In this test mode, the predistorted signal is computed using fixed-point precision. Note that the PA characteristic is assumed to be varying very slowly. Thus, the PA model is not updated continuously with every incoming data sample. The identification algorithm determines the PA model in a block-based manner. In the real-time test mode, the PA model is determined with the first block of IO data. In practice, the PA model can be updated with another block of IO data whenever changes in the PA characteristic are detected, for instance, due to aging or sudden changes of operation mode (e.g., a new channel is added in multichannel applications). The FPGA implementation of the Secant-Volterra PD and the real-time test results are presented in Section 7.

5.3. Limitations of the testbed

The testbed poses certain limitations in measurement of the nonlinear PA characteristics due to the imperfection of the available RF hardware.

As the up-converter and down-converter are nonlinear devices, the power level of the signals before these devices has to be attenuated. As a result, a low output signal level is obtained. Thus, after up-conversion and down-conversion preamplification is necessary to boost the signal to a suffi-

cient level to drive the test PA and for the signal to cover a meaningful range of the ADC, respectively. However, the preamplification increases the measurement noise floor. The increased noise floor results in a smaller dynamic range, that is, approximately 50 dB, as compared to 60 dB when measurement is done before the down-converter. This is evident in the measurements of the signal spectrum which are presented in the following two sections.

Another issue is due to the filters of the up-converter and down-converter which are bandlimited to 20 MHz. In order to model up to the fifth-order intermodulation distortion (IMD), the excitation signal bandwidth is limited to under 4 MHz. In this work, the excitation signal used is a multisine signal with 5 MHz bandwidth. Thus, the setup can only fully capture up to the third-order IMD caused by the PA.

6. THE OFFLINE TEST

The linearisation performance of the SCPWL PD and the secant Volterra PD are evaluated in the offline mode. Two test cases were considered. First, the PA is driven to a mildly nonlinear region where only third-order IMD is observed at the output spectrum, that is, with sufficient IBO. In the second test case, the PA is driven further into the nonlinear region. The results of these two test cases are presented in the following two subsections.

6.1. Results: mildly nonlinear PA

In this test, the SCPWL PD employed ten PWL partitions while the secant Volterra PD used a third-order power series as in (29) to model the PA, and the PD output is obtained by three iterations of (20).

Figure 7 shows the compensation results for the weakly nonlinear PA. The spectrum is measured after the down-converter at 70 MHz centre frequency. For comparison, an IBO was imposed on the uncompensated PA so that the in-band power of the signal is leveled to that of the compensated output. Results show that both the SCPWL PD and the secant

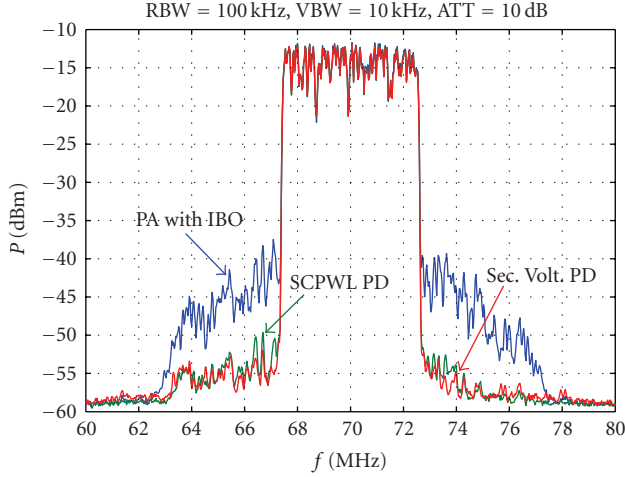


FIGURE 7: Measured power spectra of a PA driven into a weakly nonlinear region, comparison of a PA with IBO, secant Volterra PD, and SCPWL PD.

Volterra PD were able to reduce the adjacent channel power by approximately 12 dB to 15 dB.

6.2. Results : strongly nonlinear PA

The SCPWL PD employed the same number of partitions, that is, ten partitions in its model for compensation of the strongly nonlinear PA. As for the secant Volterra PD, a third-order polynomial was not sufficient for modeling the stronger nonlinearity of the PA in this case. Instead, a fifth-order power series was used to model the PA. In this test, the spectrum analyzer was placed before the down-converter so that a larger dynamic range can be observed (cf. Section 5.3).

The performance of the two PDs in the strongly nonlinear case is shown in Figure 8. The secant Volterra PD achieves an ACPR improvement of approximately 10 dB compared to 12 dB improvement in the weakly nonlinear case. The SCPWL PD outperforms the secant Volterra PD by approximately 5 dB at the best case, resulting in an ACPR reduction of 15 dB. These results may be explained by the numerical problem posed by the higher-order polynomial which leads to inaccurate modeling of the stronger compressive behaviour. In this case, a piecewise linear function offers better numerical properties for least-squares fitting.

Note that the PDs are ineffective outside of the 20 MHz mask (marked by the dashed line) of the down-converter filter since the PDs are modeled from the bandlimited IO data (i.e., IMD of fifth order and above cannot be compensated).

A relatively large IBO of 3 dB is necessary to level the in-band power of the uncompensated PA to that of the compensated ones.

7. FPGA IMPLEMENTATION AND REAL-TIME TEST

The real-time test was only performed on the iterative secant-Volterra PD presented in Section 4.2.1. In this test mode, the PD has to be first implemented on an FPGA. The implemen-

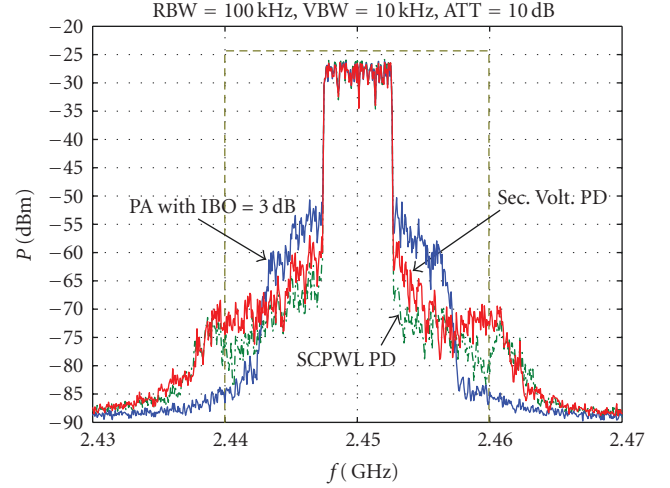


FIGURE 8: Measured power spectra of a PA driven into stronger non-linear region, comparison of a PA with IBO, secant Volterra PD, and SCPWL PD.

tation design is intended for demonstrating the implementation feasibility of the PD algorithm. Therefore, the complexity is intentionally kept minimal, where only the AM/AM characteristic of the PA is considered and is modeled using a simple Taylor series with two coefficients.

In the following subsection, the implementation of the iterative secant Volterra PD on the FPGA is described. The resource optimisation for the FPGA implementation and the fixed-point error analysis are performed before the actual implementation on the FPGA and are discussed in Section 7.2. The real-time test results are presented in Section 7.3.

7.1. FPGA implementation of the secant Volterra PD

In the implementation design, the PA is modeled with a Taylor series with first and third-order coefficients, given as

$$\begin{aligned} y[n] &= \mathbb{N}(z[n]) = \theta_1 z[n] + \theta_3 z[n] |z[n]|^2 \\ &= (\theta_1 |z[n]| + \theta_3 |z[n]|^3) e^{j \arg(z[n])}, \end{aligned} \quad (29)$$

where $z[n]$ and $y[n]$ are the input and output signal of the PA, respectively. Only two real-valued model parameters have to be estimated. It is clear that only third-order IMD products can be captured with this PA model. The two parameters θ_1 and θ_3 , along with the intended linear gain g are determined in the modeling part performed in the host computer using a MATLAB program. These parameters are needed as input to the FPGA.

Figure 9 illustrates the implementation of one iteration of the secant Volterra PD algorithm in (20). This iterative algorithm determines the output signal $z[n]$ of the secant Volterra PD. Note that in our implementation, the computation of $\mathbb{N}(z[n])$ is embedded in the function $\mathbb{T}(z[n])$. The calculation requires the PA model parameters θ_1 and θ_3 , the intended linear gain g , and the PD input signal $u[n]$ obtained from the modeling part. The required division in the

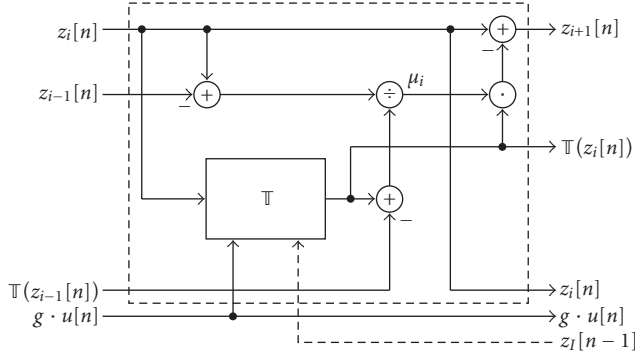


FIGURE 9: One iteration of the secant Volterra PD in detail.

algorithm is approximated with the Newton-Raphson iterative procedure in order to keep the complexity as low as possible. The details of this division algorithm are given in the appendix.

Figure 10 shows a graphical illustration of three iterations of the PD algorithm implemented on the FPGA. The first stage of the iteration starts with the two initial values $z_0[n] = u[n]$ and $z_{-1}[n] = 0$. The signal $T(z_{-1}[n]) = -g \cdot u[n]$ since $N(z_{-1}[n]) = 0$. As the product $g \cdot u[n]$ is already determined for each iteration, the initial value of $T(z_{-1}[n]) = T(0)$ requires effectively only a sign change. The following two stages require the output signal and the function T calculated from their previous stages together with the product $g \cdot u[n]$. The dashed line shows the feedback path which has to be implemented if PA models with memory are considered (not done in this implementation).

7.2. Fixed-point error analysis and resource optimization

The FPGA used in our implementation is the XC2V1000 Xilinx Virtex-2 FPGA [43]. The Xilinx Virtex-2 provides a total of forty multipliers which are implemented as hard macros.² These multipliers are optimized with respect to power consumption and speed. Therefore, the device is suitable for designs that require high clock rates, for example, algorithms that process signals with large bandwidths. The maximum bit width of these multipliers is 17 bits for unsigned values. In this design, 17 bits are used and the algorithm calculates the sign separately. Before the PD algorithm is implemented on the FPGA, the algorithm performed with fixed-point arithmetic is simulated for fixed-point error analysis. The algorithm needs to be optimized to obtain a balance between the fixed-point error and the usage of the limited resources (number of multipliers) provided by the FPGA.

At a glance from Figure 9, each iteration of the algorithm in (20) requires nine multiplications, in which three are needed for the implementation of the divider. However, the product $g \cdot u[n]$ in the function $T_u(z[n])$ need only to be

calculated in the initial stage as discussed in the last subsection. Therefore, after the initial stage, each iteration requires eight multiplications. With the forty multipliers, a maximum of four iterations can be accommodated.

Next, a Simulink model of the algorithm is implemented with 17-bit operands and fixed-point arithmetic. The output signal is compared to that generated by the same algorithm executed with floating-point double-precision arithmetic. With a multitone test signal, a third-order PA model as in (29), and with three iterations of the secant algorithm (20), the maximum relative error between the calculated signals in fixed-point precision and floating-point precision is only 1.7% [45].

Finally, three iterations of the algorithm are implemented in VHDL. A final VHDL simulation using ModelSim is performed before implementation on the FPGA [45]. The simulation provides a cycle-true and bit-true computation of the predistorted signal. Figure 11 shows a measurement result on the mildly nonlinear PA (cf. also Figure 7). The PA was excited with predistorted signals calculated in Matlab and the ModelSim simulation of the VHDL description. No performance loss due to the fixed-point error can be observed from the results.

7.2.1. FPGA resources

The developed PD design can be clocked with a maximum clock frequency of 133 MHz. Approximately 50% of the FPGA resources are used in the above implementation. The remaining resources can be used for further enhancements, for example, to support PA models with memory and/or PA models with higher-order nonlinear terms.

7.3. Measurement results: real-time test

The secant Volterra PD which was implemented on the FPGA as presented in Section 7.1 is tested in the real-time mode. Each input sample is predistorted by the PD in real-time. In this test, the PA is driven into a mildly nonlinear region where significant third-order IMD is observed, but fifth-order IMD is not significant.

The linearisation performance of the real-time secant Volterra PD is compared to that of the offline secant Volterra PD which was implemented in MATLAB (floating-point precision). Figure 12 shows the measurement results. No significant performance loss can be observed in the real-time FPGA implementation. Both the offline PD and real-time PD show excellent linearisation performance—an ACPR suppression of up to 15 dB is achieved.

The power loss in terms of required power back-off of an uncompensated PA is demonstrated in Figure 13. The uncompensated PA is backed off to achieve an equal ACPR as the compensated PA. A large IBO of 9 dB is necessary to reduce the ACI to the same level as achieved with the PD, leading to a significant in-band power loss of approximately 8 dB compared to the in-band power of the linearised PA. This proves the efficacy of the implemented PD design.

² Hard macros are unchangeable parts of programmable logic devices.

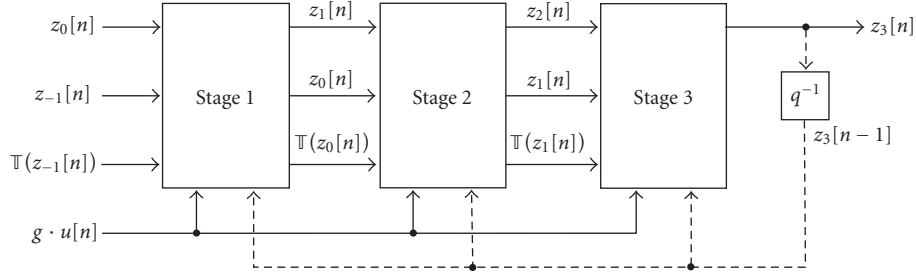


FIGURE 10: Implemented three stages of the secant Volterra PD.

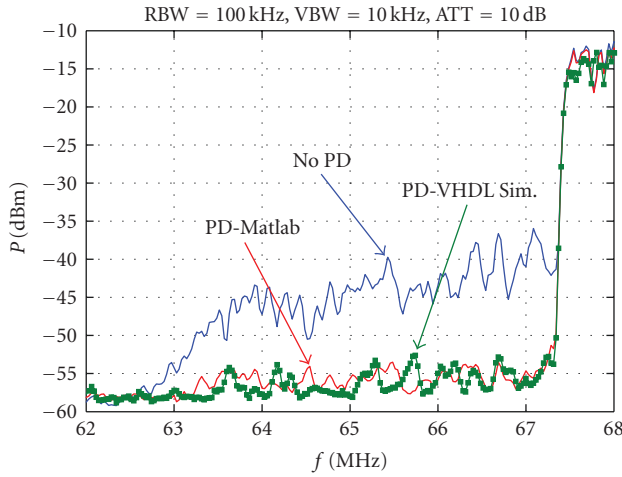


FIGURE 11: Measurement result on a mildly nonlinear PA with and without PD: PD signal is calculated with floating-point precision (PD-MATLAB) and 17 bit fixed-point precision in a ModelSim VHDL simulation (PD-VHDL Sim.).

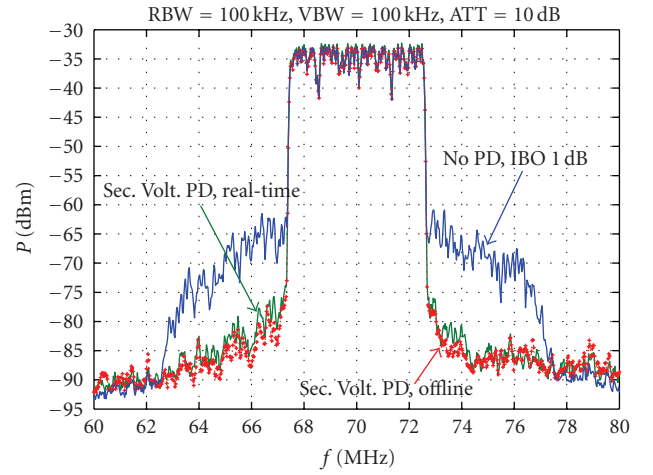


FIGURE 12: Measured output spectra at 70 MHz IF: comparison of IBO and digital predistortion (secant Volterra PD) in offline and real-time modes.

8. CONCLUSIONS

We have proposed two digital predistorters (PD) that are identified from measurement data of a broadband power amplifier (PA). A measurement testbed was built for rapid prototyping of the proposed PDs. The first PD is based on the simplicial canonical piecewise linear (SCPWL) function which is capable only of compensating amplitude-to-amplitude (AM/AM) distortion. The second PD uses a Volterra model for modeling the nonlinearities, offering the possibility to include memory effect compensation. The SCPWL-PD is identified using a least-squares (LS)-based algorithm. Due to the linear affine property of the function, the computational complexity of the identification algorithm is significantly reduced. As for the Volterra model PD, the pre-inverse model is difficult to identify. Therefore, an iterative method, namely, the secant method for root-finding, is used for the identification of the Volterra model PD.

Two test modes were set up for the proposed PDs, namely, the offline mode and the real-time mode. In the offline test mode, the PDs are identified in a host computer using the identification algorithms programmed in MATLAB. Then the excitation signal is predistorted in the host computer and transferred to the memory for transmission again.

This mode allows quick assessment of the PD performance. Both the SCPWL-PD and the Volterra PD are tested in this mode. The performance of the two PDs were evaluated on a mildly nonlinear PA and a strongly nonlinear PA. The mildly nonlinear PA exhibits only third-order intermodulation distortion (IMD) while the latter exhibits mild fifth-order IMD. It is observed that the SCPWL-PD performs better in the strong nonlinear case. This result reflects the numerical instability that polynomial models pose when modeling strong nonlinearity. Modelling inaccuracy leads to PD performance loss.

In the real-time test mode, the Volterra model PD identified using the secant method was implemented on a fixed-point arithmetic FPGA Xilinx Virtex-2 XC2V1000. In order to evaluate the implementation feasibility of the iterative method, the complexity of the model is kept minimal. A memoryless third-order power series was used and three iterations of the secant method were implemented on the FPGA. Only 50% of the FPGA resources were used in this implementation. Besides implementation feasibility and performance evaluation, this test mode also allows to compare the performance of fixed-point arithmetic and floating-point arithmetic for PD implementation. No significant performance loss in terms of adjacent channel power ratio (ACPR) is observed in the fixed-point arithmetic implementation as

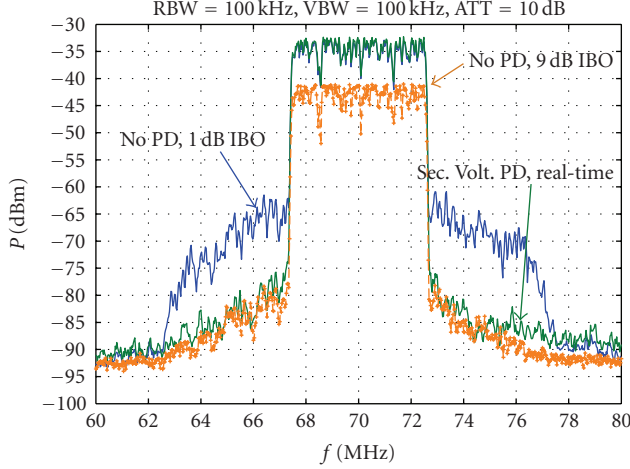


FIGURE 13: Measured output spectra at 70 MHz IF: comparison of IBO and digital predistortion with the secant Volterra PD (real-time) for achieving equal out-of-band distortions.

TABLE 1: Starting values for the Newton-Raphson method applied for performing a division $1/d$, d being represented by four bits and interpreted as a fractional number.

k	I_k	Exact value, $x = 1/d$	Starting value, $x_0 = 2^{k-1}$
1	$\left[\frac{5}{8}, 1\right]$	$\left[\frac{8}{5}, 1\right]$	$2^0 = 1$
2	$\left[\frac{3}{8}, \frac{4}{8}\right]$	$\left[\frac{8}{3}, 2\right]$	$2^1 = 2$
3	$\frac{2}{8}$	4	$2^2 = 4$
4	$\frac{1}{8}$	8	$2^3 = 8$

compared to the floating-point arithmetic implementation (MATLAB program).

Overall, both the PDs show good linearisation performance. In compensating the mildly nonlinear PA, both the PDs were able to reduce the ACPR by approximately 15 dB with the Volterra PD performing slightly better. However, when the PA is driven to a stronger nonlinear region, the performance of the Volterra model PD degraded by approximately 5 dB leading to an ACPR reduction of 10 dB while the performance of the SCPWL-PD remains the same. We have also shown that in order for the uncompensated PA to match the ACPR level of the compensated PA output, an IBO of 9 dB is required leading to an in-band power loss of 8 dB in the transmitted signal. This in turn indicates the power efficiency to gain a PD can provide for systems that require linear transmission.

APPENDIX

A. APPROXIMATION OF THE DIVISION

The FPGA provides optimised hardware multipliers but does not provide optimised hardware dividers. The XILINX Logi-

Core library provides an IP-core for a divider implementation [46] but it proves to be too costly in terms of resources. Therefore, an alternative method, based on the Newton-Raphson root-finding algorithm, is used [45]. If a division

$$r = \frac{n}{d} = n \cdot \frac{1}{d} = n \cdot x \quad (\text{A.1})$$

has to be performed, the task is to calculate $x = 1/d$ and multiply the result with the numerator n . Rearranging terms gives

$$d - \frac{1}{x} = f(x) = 0, \quad (\text{A.2})$$

which can be solved with the Newton-Raphson method

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} = x_i(2 - dx_i), \quad i \geq 0, x_0 \text{ given.} \quad (\text{A.3})$$

The convergence rate of the Newton-Raphson algorithm is quadratic, therefore, it can be expected that few iterations are sufficient. Further, the starting value x_0 can be chosen freely and, thus, a list of optimised starting values can be produced. Based on the value of d , the optimal value x_0 can be chosen. If x_0 is further chosen to be a power of two, the multiplications with x_0 reduce to cheap shift-operations. In this way, the first iteration x_1 is computed without a multiplication.

The range of the possible values for fractional numbers,³ which are used in this design, is divided into $N - 1$ intervals $I_k \equiv [2^{-k} + \Delta; 2^{-(k-1)}]$, $k = 1, 2, \dots, N - 1$, Δ being the resolution $\Delta = 2^{-(N-1)}$. The starting-value x_0 for each interval is then chosen to be $x_0 = 2^{k-1}$ if $d \in I_k$, thus, at the upper limit of the interval, the correct result is obtained with the starting value. Table 1 shows an example list of starting values, assuming that the number d is given by a fractional 1.3 two-complement representation and only positive values, ranging from 1 to Δ are taken into account. The resolution (or numerical value of the least significant bit) in this case is $\Delta = 2^{-3} = 1/8$.

It can be shown that with these starting-values the Newton-Raphson algorithm is guaranteed to converge [48]. An error analysis [48] shows that after the second iteration, the relative error $\varepsilon_2 = (x_2 - x)/x$ is only 6.25%. The arithmetic cost for the division, if two iterations are performed, is only two multiplications (the multiplications with the initial value in the first iteration are shift operations) and three subtractions. With the multiplication of the numerator, three multiplications in total are necessary.

REFERENCES

- [1] R. D. J. van Nee and R. Prasad, *OFDM for Wireless Multimedia Communications*, Artech House, London, UK, 2000.
- [2] A. R. S. Bahai, B. R. Saltzberg, and M. Ergen, *Multi-Carrier Digital Communications: Theory and Applications of OFDM*, Springer, New York, NY, USA, 2nd edition, 2004.

³ A number x can be represented with N bits in I.Q-format, $I = 1, Q = N - 1$ as [47] $x = -b_{N-1} + \sum_{k=1}^{N-1} b_{N-1-k} 2^{-k}$, $b_{N-1-k} \in \{0, 1\}$ and $-1 \leq x \leq 1 - 2^{-N+1}$. The resolution is $\Delta = 2^{-N+1}$. If $N = 4$, $\Delta = 2^{-3} = 0.125$.

- [3] "3rd Generation Partnership Project," <http://www.3gpp.org>.
- [4] P. B. Kenington, *High Linearity RF Amplifier Design*, Artech House, London, UK, 2000.
- [5] T. Nojima, T. Murase, and N. Imai, "The design of a predistortion linearization circuit for high-level modulation radio systems," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '85)*, vol. 3, pp. 1466–1471, New Orleans, La, USA, December 1985.
- [6] T. Nojima and T. Konno, "Cuber predistortion linearizer for relay equipment in 800 MHz band land mobile telephone system," *IEEE Transactions on Vehicular Technology*, vol. 34, no. 4, pp. 169–177, 1985.
- [7] S. P. Stapleton and F. C. Costescu, "An adaptive predistorter for a power amplifier based on adjacent channel emissions," *IEEE Transactions on Vehicular Technology*, vol. 41, no. 1, pp. 49–56, 1992.
- [8] A. Bateman, D. M. Haines, and R. J. Wilkinson, "Linear transceiver architectures," in *Proceedings of the 38th IEEE Vehicular Technology Conference (VTC '88)*, pp. 478–484, Philadelphia, Pa, USA, June 1988.
- [9] M. Faulkner and M. Johansson, "Adaptive linearization using predistortion-experimental results," *IEEE Transactions on Vehicular Technology*, vol. 43, no. 2, pp. 323–332, 1994.
- [10] J. K. Cavers, "A linearizing predistorter with fast adaptation," in *Proceedings of the 40th IEEE Vehicular Technology Conference (VTC '90)*, pp. 41–47, Orlando, Fla, USA, May 1990.
- [11] E. Changsoo and E. J. Powers, "A new volterra predistorter based on the indirect learning architecture," *IEEE Transactions on Signal Processing*, vol. 45, no. 1, pp. 223–227, 1997.
- [12] A. Saleh, "Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers," *IEEE Transactions on Communications*, vol. 29, no. 11, pp. 1715–1720, 1981.
- [13] Y. Nagata, "Linear amplification technique for digital mobile communications," in *Proceedings of the 39th IEEE Vehicular Technology Conference (VTC '89)*, pp. 159–164, San Francisco, Calif, USA, May 1989.
- [14] E. G. Jeckeln, F. M. Ghannouchi, and M. A. Sawan, "An L band adaptive digital predistorter for power amplifiers using direct I-Q modem," in *Proceedings of IEEE MTT-S International Microwave Symposium Digest (MWSYM '98)*, vol. 2, pp. 719–722, Baltimore, Md, USA, June 1998.
- [15] S. Boumaiza, J. Li, and F. M. Ghannouchi, "Implementation of an adaptive digital/RF predistorter using direct LUT synthesis," in *Proceedings of IEEE MTT-S International Microwave Symposium (IMS '04)*, vol. 2, pp. 681–684, Fort Worth, Tex, USA, June 2004.
- [16] H. Ben Nasr, S. Boumaiza, M. Helaoui, A. Ghazel, and F. M. Ghannouchi, "On the critical issues of DSP/FPGA mixed digital predistorter implementation," in *Proceedings of Asia-Pacific Conference on Microwave Conference (APMC '05)*, vol. 5, p. 4, Suzhou, China, December 2005.
- [17] L. Ding, H. Qian, N. Chen, and G. T. Zhou, "A memory polynomial predistorter implemented using TMS320C67xx," in *Proceedings of Texas Instruments Developer Conference*, pp. 1–7, Houston, Tex, USA, February 2004.
- [18] N. Chen, G. T. Zhou, and H. Qian, "Power efficiency improvements through peak-to-average power ratio reduction and power amplifier linearization," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, Article ID 20463, 7 pages, 2007.
- [19] L. Ding, Z. Ma, D. R. Morgan, M. Zierdt, and J. Pastalan, "A least-squares/Newton method for digital predistortion of wideband signals," *IEEE Transactions on Communications*, vol. 54, no. 5, pp. 833–840, 2006.
- [20] "Mini-Circuits ZVE-8G Amplifier," <http://www.minicircuits.com/pdfs/ZVE-8G.pdf>.
- [21] 3rd Generation Partnership Project, "Technical Specification Group Radio Access Network; Base Station (BS) radio transmission and reception (FDD) (Release 6), TS 25.104," <http://www.3gpp.org>.
- [22] F. H. Raab, P. Asbeck, S. Cripps, et al., "Power amplifiers and transmitters for RF and microwave," *IEEE Transactions on Microwave Theory and Techniques*, vol. 50, no. 3, pp. 814–826, 2002.
- [23] L. O. Chua and S. M. Kang, "Section-wise piecewise-linear functions: canonical representation, properties, and applications," *Proceedings of the IEEE*, vol. 65, no. 6, pp. 915–929, 1977.
- [24] P. Julian, A. Desages, and O. Agamennoni, "High-level canonical piecewise linear representation using a simplicial partition," *IEEE Transactions on Circuits and Systems*, vol. 46, no. 4, pp. 463–480, 1999.
- [25] M.-J. Chien and E. Kuh, "Solving nonlinear resistive networks using piecewise-linear analysis and simplicial subdivision," *IEEE Transactions on Circuits and Systems*, vol. 24, no. 6, pp. 305–317, 1977.
- [26] C. Guzelis and I. C. Goknar, "A canonical representation for piecewise-affine maps and its applications to circuit analysis," *IEEE Transactions on Circuits and Systems*, vol. 38, no. 11, pp. 1342–1354, 1991.
- [27] B. Fronberg, *A Practical Guide to Pseudospectral Methods*, Cambridge University Press, New York, NY, USA, 1998.
- [28] M. Y. Cheong, S. Werner, J. Cousséau, J. L. Figueroa, and T. I. Laakso, "Predistorter design employing parallel piecewise linear structure and inverse coordinate mapping for broadband communications," in *Proceedings of the 14th European Signal Processing Conference (EUSIPCO '06)*, pp. 1–5, Florence, Italy, September 2006.
- [29] N. Wiener, *Nonlinear Problems in Random Theory*, John Wiley & Sons, New York, NY, USA, 1958.
- [30] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*, John Wiley & Sons, New York, NY, USA, 1980.
- [31] S. Boyd and L. O. Chua, "Fading memory and the problem of approximating nonlinear operators with volterra series," *IEEE Transactions on Circuits and Systems*, vol. 32, no. 11, pp. 1150–1161, 1985.
- [32] D. Hummels and R. Gitchell, "Equivalent low-pass representations for bandpass volterra systems," *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 140–142, 1980.
- [33] M. Y. Cheong, E. Aschbacher, P. Brunmayr, M. Rupp, and T. I. Laakso, "Comparison and experimental verification of two low-complexity digital predistortion methods," in *Proceedings of the 39th Asilomar Conference on Signals, Systems and Computers*, pp. 432–436, Pacific Grove, Calif, USA, October–November 2005.
- [34] M. Schetzen, "Theory of pth-order inverses of nonlinear systems," *IEEE Transactions on Circuits and Systems*, vol. 23, no. 5, pp. 285–291, 1976.
- [35] D. G. Luenberger, *Optimization by Vector Space Methods*, John Wiley & Sons, New York, NY, USA, 1968.
- [36] A. M. Ostrowski, *Solution of Equations and Systems of Equations*, Academic Press, New York, NY, USA, 1960.
- [37] R. D. Nowak and B. D. V. Veen, "Volterra filter equalization: a fixed point approach," *IEEE Transactions on Signal Processing*, vol. 45, no. 2, pp. 377–388, 1997.
- [38] Advanced Design System (ADS), "Agilent Technologies," http://eesof.tm.agilent.com/products/ads_main.html.

- [39] *Sundance SMT310Q PCI Carrier Board*, <http://www.sundance.com/docs/SMT310Q%20User%20Manual.pdf>.
- [40] *Sundance SMT351-G module*, <http://www.sundance.com/docs/SMT351%20User%20Manual.pdf>.
- [41] *Sundance SMT370-AC module*, <http://www.sundance.com/docs/SMT370%20User%20Manual.pdf>.
- [42] *Analog Devices D/A converter AD9777*, http://www.analog.com/UploadedFiles/Data_Sheets/3229938536490156500AD-9777.b.pdf.
- [43] *XILINX Virtex-II Platform FPGA*, <http://www.xilinx.com/partinfo/ds031.pdf>.
- [44] *Sundance SMT365 module*, <http://www.sundance.com/docs/smt365%20user%20manual.pdf>.
- [45] P. Brunmayr, "Implementation of a nonlinear digital pre-distortion algorithm," M.S. thesis, Vienna University of Technology, Institute of Communications and Radio-Frequency Engineering, Vienna, Austria, 2005, http://publik.tuwien.ac.at/files/pub-et_10048.pdf.
- [46] *XILINX IP Core Pipelined Divider*, http://www.xilinx.com/bvdocs/ipcenter/data_sheet/sdivider.pdf.
- [47] G. Doblinger, *Signalprozessoren—Architekturen, Algorithmen, Anwendungen*, J. Schlembach Fachverlag, Weil der Stadt, Germany, 2000.
- [48] E. Aschbacher, "Digital pre-distortion of microwave power amplifiers," Ph.D. dissertation, Vienna University of Technology, Institute of Communications and Radio-Frequency Engineering, Austria, 2005, http://publik.tuwien.ac.at/files/pub-et_11772.pdf.