Minimum Mean-Square Error Equalization for Second-Order Volterra Systems

Christoph Krall, *Member, IEEE*, Klaus Witrisal, *Member, IEEE*, Geert Leus, *Senior Member, IEEE*, and Heinz Koeppl

***Abstract*—In this paper, a novel nonlinear Volterra equalizer is presented. We define a framework for nonlinear second-order Volterra models, which is applicable to different applications in engineering. We use this framework to define the channel and to model the equalizer. We then solve the minimum mean-square error (MMSE) problem explicitly for the tandem connection of the two second-order Volterra systems. Optimal solutions for a simplified, linear version of the MMSE equalizer are also presented. The novel equalizer was tested when applied to a nonlinear ultra-wide-band transmitted reference receiver front end. As a comparison, a least mean squares (LMS) equalizer with a training sequence has been used to verify the performance of the newly proposed equalizer. The simulation results show that the LMS equalizer is only able to attain the proposed MMSE equalizer after very long training, which might not be desirable in a communication system.**

***Index Terms*—Nonlinear MMSE equalizer, second-order systems, Volterra systems.**

I. Introduction

NONLINEARITIES are encountered in many communication systems. This results from the fact that systems are pushed to their limits, thus linear modeling of the components is not sufficient any more to achieve a good model accuracy. As the information bearing signals are processed and distorted by nonlinearities, there is a high interest to find algorithms that reconstruct this information. Thus, the problems occurring due to nonlinearities in the analog domain are compensated for by means of nonlinear digital signal processing. Most of the time, the nonlinearities encountered in the continuous time domain are static nonlinearities, which means that they do not show any memory effects and a set of input values is mapped by a nonlinear function to a set of output values. This function can be known exactly by an analytical expression or can be approximated by means of Taylor series, neural networks [1], and many other modeling approaches. If the memory is included in the model, one can develop equivalent nonlinear fading memory systems and use Volterra models to describe the input–output behavior [2], [3]. This approach is used in many applications for nonlinear system description. Examples of nonlinear Volterra system modeling are presented in [4] where nonlinear magnetic recording channels are modeled, in [5] nonlinear digital satellite channels are modeled, and in [6] the authors have shown that nonlinear Volterra modeling is also possible for noncoherent ultra-wide-band (UWB) receiver front ends. Many more applications can be found where Volterra modeling is successfully applied, see, e.g., [7]–[9].

To recover the originally transmitted information, an equalizer has to be used to combat the distortion effects of the channel. For strictly linear systems, many different equalization strategies may be used. The most commonly known are zero-forcing (ZF) equalizers, minimum mean-square error (MMSE) equalizers, maximum-likelihood (ML) equalizers, and minimum bit error rate (MBER) equalizers [10]. For linear finite-impulse-response (FIR) channels, the definition of an exact inverse is given by replacing all the zeros of the channel transfer function by poles, thus leading to infinite-impulse-response (IIR) filters. A minimum phase requirement on the channel is thus implicitly assumed to be able to create a stable and causal ZF equalizer [11]. Any occurring noise in the overall system is amplified by a possible gain of the ZF equalizer, making it useless for many implementations.

Another possibility to compute an equalizer is to minimize the mean squared error over a block of data symbols. Usually the error is then defined as the difference between the originally transmitted data symbols and the estimated data symbols, i.e., the received data symbols processed by an equalizer. Thus, the formulation of the overall system is necessary, i.e., the combined response of the channel and the equalizer. For purely linear systems, the MMSE equalizer is easily derived and can be found in textbooks [12], [13].

In this paper, we extend the conventional linear MMSE equalizer to nonlinear, second-order Volterra systems. Generally, nonlinear fading memory systems are described by a set of kernels, i.e., characterizing the nonlinear input-output behavior of the system. The output of a discrete-time, qth-order Volterra system with input $d[k]$ is given by

$$\begin{aligned} z[k] &= h_0 + \sum_{n_1=0}^{N_1-1} h_1[n_1] d[k-n_1] \\ &+ \sum_{n_1=0}^{N_2-1}\sum_{n_2=0}^{N_2-1} h_2[n_1,n_2] d[k-n_1] d[k-n_2] \\ &+ \cdots + \sum_{n_1=0}^{N_q-1}\cdots\sum_{n_q=0}^{N_q-1} h_q[n_1,\ldots,n_q]\prod_{i=1}^{q} d[k-n_i] \end{aligned} \tag{1}$$

Manuscript received September 3, 2007; revised May 13, 2008. First published July 9, 2008; current version published September 17, 2008. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Manuel Davy. C. Krall would like to thank the Austrian Research Centers (ARC) GmbH, Austria, and the Christian Doppler Laboratory for Nonlinear Signal Processing for funding. G. Leus is supported in part by NWO-STW under the VIDI program (DCT.6577).

C. Krall is with the Radiocommunications Systems Division, Rohde & Schwarz GmbH & Company KG, D-81671 Munich, Germany (e-mail: christoph.krall@ieee.org).

K. Witrisal is with the Signal Processing and Speech Communication Laboratory, Graz University of Technology, A-8010 Graz, Austria (e-mail: witrisal@tugraz.at).

G. Leus is with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2826 CD Delft, The Netherlands (e-mail: g.j.t.leus@tudelft.nl).

H. Koeppl is with the Laboratory of Nonlinear Systems (LANOS), Ecole Polytechnique Federale de Lausanne, CH-1015 Lausanne, Switzerland (e-mail: Heinz.koeppl@epfl.ch).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2008.928167

where h_0 and h_1 represent the bias and linear part, respectively. All higher-order kernels are given as h_i, $i = 2, \ldots, q$. Each of the multidimensional objects h_i has, in its most general form, independent memory depths N_i, $i = 1, \ldots, q$. The data sequence $d[k]$ represents data symbols with an arbitrary modulation format.

Generally, an equalizer for a nonlinear Volterra system can be computed exactly as a recursive Volterra system as shown in [14]. However, due to the recursive structure, the inverse system may encounter stability problems when applying signals with low signal-to-noise ratio (SNR) [15]. Conversely to the exact inverse, a ZF Volterra equalizer can be found by expressing the cascade of the Volterra channel and Volterra equalizer in terms of the orders of the input signal and equating equal exponents. This is conventionally known as the pth-order inverse of nonlinear systems [16]. Up to an order of p all the nonlinear kernels of the cascade are forced to be zero. By introducing a nonlinear equalizer, higher-order terms (larger than p) occur in the cascade, which are nonzero and are assumed to be small such that their influence is negligible. However, exact inverses as well as pth-order inverses require that the first-order kernel of the channel is minimum-phase because an exact inverse of the linear part is needed.

In this paper, we want to address an exact expression for an MMSE equalizer of a second-order Volterra system. To realize this, we make the crucial assumption that the data symbols $d[k]$ in (1) are binary antipodal with zero mean. Under this assumption, we extend the linear MMSE equalizer by incorporating the second-order Volterra channel model, and we adapt the MSE formulation to second-order Volterra equalizers in order to improve the performance even further. Additionally, the obtained results are compared with their least mean squares (LMS) counterparts, which ideally should achieve similar results. Note that initial work on these approaches has been reported in [17], but the equalizers developed there are only approximations of the true MMSE equalizers. In [18], linear equalizers are proposed to achieve equalization of a nonlinear Volterra system. Such an equalizer requires oversampling at the receiver front end increasing the complexity significantly. Still, it has been applied successfully to oversampled nonlinear receiver front ends in [19]. In other works, the equalization problem has been formulated as a fixed point problem [20], [21] where a solution for the equalizer is found iteratively. This is possible as long as the mapping between the iteration steps is contractive. An effective implementation of an adaptive equalizer in the frequency-domain is found in [22] where fast block convolution algorithms are used to combat intersymbol interference (ISI). In [23] a nonlinear least squares (LS) equalizer is found for IIR nonlinear systems and the authors in [24] propose an iterative method with a nonlinear predictor.

The paper is organized as follows. In Section II, the nonlinear channel for our approach is defined. A similarly defined generalized equalizer structure is shown in Section III. We express the MMSE solutions for linear and second-order Volterra equalizers in Section IV. Section V briefly reviews the adaptive nonlinear filter definitions achieving MMSE equalization by LMS. Simulation results for an example of such a nonlinear second-order Volterra system, fitting in the proposed framework, is presented in Section VI. Finally, conclusions are drawn in Section VII.

II. Nonlinear System Model

As seen in (1), the output of a fading-memory nonlinear system can be considered as a generalization of the linear convolution. This extension is obtained by including nonlinear terms in this convolution, i.e., by using all product terms for different time lags up to a certain memory depth. For the special case of a second-order system, i.e., $h_q = 0, \forall q > 2$, we can write the output of the Volterra system in a matrix-vector form as

$$z[k] = h_0 + \mathbf{h}_1^{\mathsf{T}}\mathbf{d}[k] + \mathbf{d}^{\mathsf{T}}[k]\mathbf{H}_2\mathbf{d}[k] + n[k] \qquad (2)$$

where $n[k]$ is the additive noise, $\mathbf{d}[k] = [d[k], d[k-1], \ldots, d[k-L+1]]^{\mathsf{T}}$ is a vector containing the L data symbols, and where the scalar h_0, the $L \times 1$ vector $\mathbf{h}_1$, and the $L \times L$ upper triangular matrix $\mathbf{H}_2$ are the zeroth-, first-, and second-order kernels of the channel, respectively. With respect to (1), the memory depth of the nonlinear system is defined as $L = \max(N_1, N_2)$. We make the following assumptions in this work:

A1) the data symbols $d[k]$ are binary antipodal, i.e., $d[k] \in \{\pm c\}$, with $c \in \mathbb{R}$ and $\pm c$ is equiprobable such that $\mathsf{E}\{d[k]\} = 0$;

A2) the noise is i.i.d. Gaussian with zero-mean and variance σ_n^2.

The derivations in this paper can also be done without the i.i.d. assumption. However, the covariance matrices introduced later on will not be scaled identity matrices anymore.

The squared elements of $\mathbf{d}[k]$ are given as $d^2[k] = c^2, \quad \forall\, k$, and thus the main diagonal of the second-order kernel $\mathbf{H}_2$ can be included into the bias term h_0 which simplifies (2).

For our further considerations, we assume that the bias term is not contained in the system and we define

$$y[k] = z[k] - h_0 = \mathbf{h}_1^{\mathsf{T}}\mathbf{d}[k] + \mathbf{d}^{\mathsf{T}}[k]\mathbf{H}_2\mathbf{d}[k] + n[k] \qquad (3)$$

as the received sequence without the bias h_0, which can easily be obtained by averaging.

Since we want to write the second-order kernel as a linear function of the crossterms, we introduce the operator $\boxtimes$, which is a modified Kronecker product that only takes the unique half of the crossterms into account. We can then rewrite $y[k]$ as

$$y[k] = \mathbf{h}_1^{\mathsf{T}}\mathbf{d}[k] + \mathbf{h}_2^{\mathsf{T}}(\mathbf{d}[k] \boxtimes \mathbf{d}[k]) + n[k] \qquad (4)$$

where $\mathbf{h}_2$ is an appropriate modification of $\mathbf{H}_2$, omitting also the main diagonal elements since they have been collected in the bias term. For a vector $\mathbf{b} = [b_1, b_2, \ldots, b_K]^{\mathsf{T}}$, we define the reduced Kronecker product $\mathbf{b} \boxtimes \mathbf{b}$ as $\mathbf{b} \boxtimes \mathbf{b} = [b_1b_2, b_2b_3, \ldots, b_{K-1}b_K, b_1b_3, \ldots, b_{K-2}b_K, \ \ldots, b_1b_{K-1}, b_2b_K, b_1b_K]^{\mathsf{T}}$.

III. Equalizer Model

We can now define an equalizer that is applied to the received signal $y[k]$. For a first-order Volterra or linear equalizer the estimated data sequence $\hat{d}[k]$ becomes

$$\hat{d}[k] = \mathbf{g}_1^{\mathsf{T}}\mathbf{y}[k] \tag{5}$$

where $\mathbf{y}[k] = [y[k], y[k-1], \ldots, y[k-L_e+1]]^{\mathsf{T}}$, and where the $L_e \times 1$ vector $\mathbf{g}_1$ represents the linear equalizer coefficients. Applying a second-order Volterra equalizer to $y[k]$, we obtain

$$\hat{d}[k] = \mathbf{g}_1^{\mathsf{T}}\mathbf{y}[k] + \mathbf{y}^{\mathsf{T}}[k]\mathbf{G}_2\mathbf{y}[k] \tag{6}$$

where the $L_e \times L_e$ matrix $\mathbf{G}_2$ represents the second-order kernel of the Volterra equalizer. Note that there is no zeroth-order kernel in both cases, since we already removed the bias from the channel model.

As we did for the channel, we can rewrite $\hat{d}[k]$ as

$$\hat{d}[k] = \mathbf{g}_1^{\mathsf{T}}\mathbf{y}[k] + \mathbf{g}_2^{\mathsf{T}}(\mathbf{y}[k] \boxtimes \mathbf{y}[k]) \tag{7}$$

where $\mathbf{g}_2$ is an appropriate modification of $\mathbf{G}_2$.

IV. MMSE Volterra Filters

In this section we derive linear and nonlinear equalizers for the nonlinear channel model.

A. First-Order Equalizer

First of all, let us assume that the second-order kernel $\mathbf{g}_2$ is zero and that we only focus on the first-order kernel $\mathbf{g}_1$. We then have to derive an expression for $\mathbf{y}[k]$ as a function of $\mathbf{h}_1$ and $\mathbf{h}_2$. To this end, let us rewrite the L_e elements of the input samples for the equalizer $y[k-l]$ as

$$y[k-l] = \mathbf{h}_{1,l}^{\mathsf{T}}\mathbf{d}_x[k] + \mathbf{h}_{2,l}^{\mathsf{T}}(\mathbf{d}_x[k] \boxtimes \mathbf{d}_x[k]) + n[k-l] \tag{8}$$

where the extended data vector is given by $\mathbf{d}_x[k] = [d[k], d[k-1], \ldots, d[k-L-L_e+2]]^{\mathsf{T}}$ and where the channel vectors $\mathbf{h}_{1,l}$ and $\mathbf{h}_{2,l}$ are appropriate extensions of $\mathbf{h}_1$ and $\mathbf{h}_2$ which depend on $l = 0, 1, \ldots, L_e - 1$, respectively.

In short, we can thus write

$$y[k-l] = \mathbf{f}_l^{\mathsf{T}}\mathbf{s}[k] + n[k-l] \tag{9}$$

where $\mathbf{f}_l = \left[\mathbf{h}_{1,l}^{\mathsf{T}}, \mathbf{h}_{2,l}^{\mathsf{T}}\right]^{\mathsf{T}}$ and $\mathbf{s}[k] = [\mathbf{d}_x^{\mathsf{T}}[k], (\mathbf{d}_x[k] \boxtimes \mathbf{d}_x[k])^{\mathsf{T}}]^{\mathsf{T}}$. Both vectors, $\mathbf{f}_l$ and $\mathbf{s}[k]$ are having the dimension $(\varepsilon + \eta) \times 1$, where $\varepsilon = L_e + L - 1$ denotes the length of the linear terms, and $\eta = \sum_{\alpha=2}^{L}(L_e + L - \alpha)$ denotes the length of the nonlinear product terms.

It is then easy to see that we can write $\mathbf{y}[k]$ as

$$\mathbf{y}[k] = \mathbf{F}\mathbf{s}[k] + \mathbf{n}[k] \tag{10}$$

$$\mathbf{F} = \left[\begin{array}{cccc}
h_1[0] & 0 & \cdots & 0 \\
h_1[1] & h_1[0] & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & \cdots & h_1[L-1] & h_1[L-2] \\
0 & \cdots & 0 & h_1[L-1] \\
\hline
h_2[0,1] & 0 & \cdots & 0 \\
h_2[1,2] & h_2[0,1] & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & \cdots & h_2[L-1,L-2] & h_2[L-2,L-3] \\
0 & \cdots & 0 & h_2[L-1,L-2] \\
\hline
h_2[0,2] & 0 & \cdots & 0 \\
h_2[1,3] & h_2[0,2] & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & \cdots & h_2[L-1,L-3] & h_2[L-2,L-4] \\
0 & \cdots & 0 & h_2[L-1,L-3] \\
\hline
\vdots & \ddots & \ddots & \vdots \\
\hline
h_2[L-1,L-1] & 0 & \cdots & 0 \\
0 & h_2[L-1,L-1] & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & \cdots & 0 & h_2[L-1,L-1]
\end{array}\right]^{\mathsf{T}}
\begin{array}{l}
\\ \\ Le+L-1 \\ \\ \\
\\ \\ Le+L-2 \\ \\ \\
\\ \\ Le+L-3 \\ \\ \\
\\
\\ Le \\ \\ \\
\end{array} \tag{11}$$

where $\mathbf{n}[k]$ is similarly defined as $\mathbf{y}[k]$, and where $\mathbf{F}$ is a structured matrix containing the entries of each $\mathbf{f}_l$, i.e., $\mathbf{F} = [\mathbf{f}_0, \mathbf{f}_1, \ldots, \mathbf{f}_{L_e-1}]^\mathsf{T}$. The explicit structure of the channel matrix is given by (11), shown at the bottom of the previous page, where $h_1[k]$ is the $(k+1)$th entry of the first-order kernel $\mathbf{h}_1$, and $h_2[k,l]$ is the $(k+1, l+1)$th entry of the second-order kernel $\mathbf{H}_2$, as used in (1). Finally, the matrix $\mathbf{F}$ is of dimension $(\varepsilon + \eta) \times L_e$. For the linear terms it has the conventional Toeplitz structure and for the nonlinear terms it has a Toeplitz structure built from off-diagonals of the second-order kernel.

For finding an equalizer, we can minimize the cost function $\mathcal{J} = \mathsf{E}\{(d[k] - \hat{d}[k])^2\}$. By incorporating the linear part of (7) and solving for the MMSE equalizer $\hat{\mathbf{g}}_1$, we obtain

$$\hat{\mathbf{g}}_1 = (\mathbf{F}\mathbf{R}_s\mathbf{F}^\mathsf{T} + \mathbf{R}_n)^{-1}\mathbf{F}\mathbf{R}_s\mathbf{e}_{\delta+1} \tag{12}$$

where $\mathbf{R}_s = \mathrm{E}\{\mathbf{s}[k]\mathbf{s}^\mathsf{T}[k]\}$, $\mathbf{e}_{\delta+1}$ is a unit column vector with a "1" in position $\delta+1$, with δ the delay of the equalizer, and $\mathbf{R}_n$ is the autocorrelation matrix of the noise. Under assumption A2), the noise correlation matrix can be written as $\mathbf{R}_n = \sigma_n^2\mathbf{I}_{L_e}$. Further, under assumption A1), we can also define the structure of the autocorrelation matrix $\mathbf{R}_s$ as

$$\mathbf{R}_s = \left[\begin{array}{c:c} c^2\mathbf{I}_\varepsilon & \mathbf{0} \\ \hdashline \mathbf{0} & c^2\mathbf{I}_\eta \end{array}\right]. \tag{13}$$

The indices in the subscript of the identity matrices denote the size of the corresponding matrix, i.e., ε for the linear samples and η for the nonlinear samples, respectively. The final result in (12) is similar to the conventional purely linear equalizer and it can be viewed as a generalization of this MMSE equation. With our assumptions A1) and A2), the correlation matrices can be computed with the method shown in Appendix II. It is only questionable whether this linear solution shows good performance. This may be true for weakly nonlinear systems only.

B. Second-Order Equalizer

What if we do not assume that the second-order kernel $\mathbf{g}_2$ is zero? We then have to look for an additional expression of $\mathbf{y}[k] \boxtimes \mathbf{y}[k]$ as a function of $\mathbf{h}_1$ and $\mathbf{h}_2$.

In general, we can write that

$$\begin{aligned}\mathbf{y}[k] \boxtimes \mathbf{y}[k] &= \mathbf{S}(\mathbf{y}[k] \otimes \mathbf{y}[k]) \\ &= \mathbf{S}[(\mathbf{F}\mathbf{s}[k] + \mathbf{n}[k]) \otimes (\mathbf{F}\mathbf{s}[k] + \mathbf{n}[k])] \\ &= \mathbf{S}(\mathbf{F} \otimes \mathbf{F})(\mathbf{s}[k] \otimes \mathbf{s}[k]) \\ &\quad + \mathbf{S}(\mathbf{F} \otimes \mathbf{I})(\mathbf{s}[k] \otimes \mathbf{n}[k]) \\ &\quad + \mathbf{S}(\mathbf{I} \otimes \mathbf{F})(\mathbf{n}[k] \otimes \mathbf{s}[k]) \\ &\quad + \mathbf{S}(\mathbf{n}[k] \otimes \mathbf{n}[k])\end{aligned} \tag{14}$$

where $\mathbf{S}$ is a selection matrix that transforms the Kronecker product $\otimes$ into the modified Kronecker product $\boxtimes$. Considering now that a commutation of the Kronecker product is achieved by multiplying it by a permutation matrix $\mathbf{P}$ (see Appendix I), we can rewrite (14) as

$$\begin{aligned}\mathbf{y}[k] \boxtimes \mathbf{y}[k] &= \mathbf{S}(\mathbf{y}[k] \otimes \mathbf{y}[k]) \\ &= \mathbf{S}(\mathbf{F} \otimes \mathbf{F})(\mathbf{s}[k] \otimes \mathbf{s}[k]) \\ &\quad + \mathbf{S}[(\mathbf{F} \otimes \mathbf{I}) + (\mathbf{I} \otimes \mathbf{F})\mathbf{P}](\mathbf{s}[k] \otimes \mathbf{n}[k]) \\ &\quad + (\mathbf{n}[k] \boxtimes \mathbf{n}[k]).\end{aligned} \tag{15}$$

The output of the second-order equalizer can be written as

$$\hat{d}[k] = \mathbf{g}_{1,2}^\mathsf{T}\mathbf{w}[k] \tag{16}$$

where $\mathbf{g}_{1,2} = [\mathbf{g}_1^\mathsf{T}, \mathbf{g}_2^\mathsf{T}]^\mathsf{T}$ and $\mathbf{w}[k] = [\mathbf{y}^\mathsf{T}[k], (\mathbf{y}[k] \boxtimes \mathbf{y}[k])^\mathsf{T}]^\mathsf{T}$.

We can now write $\mathbf{w}[k]$ as

$$\mathbf{w}[k] = \mathbf{Q}\mathbf{r}[k] + \mathbf{U}(\mathbf{s}[k] \otimes \mathbf{n}[k]) + \mathbf{m}[k] \tag{17}$$

where $\mathbf{m}[k] = [\mathbf{n}^\mathsf{T}[k], (\mathbf{n}[k] \boxtimes \mathbf{n}[k])^\mathsf{T}]^\mathsf{T}$, i.e., $\mathbf{m}[k]$ is similarly defined as $\mathbf{w}[k]$, $\mathbf{r}[k] = [\mathbf{s}^\mathsf{T}[k], (\mathbf{s}[k] \otimes \mathbf{s}[k])^\mathsf{T}]^\mathsf{T}$, and $\mathbf{U} = [\mathbf{0}^\mathsf{T}, [\mathbf{S}[(\mathbf{F} \otimes \mathbf{I}) + (\mathbf{I} \otimes \mathbf{F})\mathbf{P}]]^\mathsf{T}]^\mathsf{T}$. The big channel matrix $\mathbf{Q}$ containing all products up to fourth order is given as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{F} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}(\mathbf{F} \otimes \mathbf{F}) \end{bmatrix}.$$

Now there is no correlation between the useful term $\mathbf{r}[k]$ and the noise terms $\mathbf{s}[k] \otimes \mathbf{n}[k]$ and $\mathbf{m}[k]$, since we've used the reduced Kronecker notations and zero-mean properties for $\mathbf{s}[k]$ and $\mathbf{m}[k]$, respectively.

To make things more clear, the final equation for $\mathbf{w}[k]$ is rewritten in a matrix form

$$\begin{aligned}\begin{bmatrix} \mathbf{y}[k] \\ \mathbf{y}[k] \boxtimes \mathbf{y}[k] \end{bmatrix} &= \begin{bmatrix} \mathbf{F} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}(\mathbf{F} \otimes \mathbf{F}) \end{bmatrix} \begin{bmatrix} \mathbf{s}[k] \\ \mathbf{s}[k] \otimes \mathbf{s}[k] \end{bmatrix} \\ &\quad + \begin{bmatrix} \mathbf{0} \\ \mathbf{S}[(\mathbf{F} \otimes \mathbf{I}) + (\mathbf{I} \otimes \mathbf{F})\mathbf{P}] \end{bmatrix} [\mathbf{s}[k] \otimes \mathbf{n}[k]] \\ &\quad + \begin{bmatrix} \mathbf{n}[k] \\ \mathbf{n}[k] \boxtimes \mathbf{n}[k] \end{bmatrix}.\end{aligned} \tag{18}$$

The formula for the MMSE expression for $\mathbf{g}_{1,2}$ is now easily derived. Let us define the correlation matrices as

$$\mathbf{R}_r = \mathsf{E}\{\mathbf{r}[k]\mathbf{r}^\mathsf{T}[k]\} \tag{19}$$

$$\mathbf{R}_{s,n} = \mathsf{E}\{[(\mathbf{s}[k] \otimes \mathbf{n}[k])][(\mathbf{s}[k] \otimes \mathbf{n}[k])]^\mathsf{T}\} \tag{20}$$

and

$$\mathbf{R}_m = \mathsf{E}\{\mathbf{m}[k]\mathbf{m}^\mathsf{T}[k]\}. \tag{21}$$

One can show that the correlation between the data vector $\mathbf{r}[k]$ and the noise vector $\mathbf{n}[k]$ is zero since the assumption that data and noise are uncorrelated even holds for this modified data and noise vector. But this property only holds as long as we assume that the second-order kernels of the channel and the equalizer have zeros on the main diagonal.

With these assumptions we can use the same cost function as before to minimize. The parameters for the optimal equalizer are computed as

$$\hat{\mathbf{g}}_{1,2} = (\mathbf{Q}\mathbf{R}_r\mathbf{Q}^\top + \mathbf{U}\mathbf{R}_{s,n}\mathbf{U}^\top + \mathbf{R}_m)^{-1}\mathbf{Q}\mathbf{R}_r\mathbf{e}_{\delta+1}. \quad (22)$$

The correlation matrix $\mathbf{R}_r$ is not easy to describe in an analytical way. We know that the very first part is given by a diagonal matrix $c^2\mathbf{I}$ because these elements describe the correlation between the purely linear symbols. The rest of the matrix represents different terms of correlation in the linear and nonlinear data parts. Due to the modified Kronecker notation used, a simplification of the correlation matrix is a very challenging task. However, we give a detailed description on how to describe the correlation matrix in Appendix II, which can be used for any correlation matrix used in this paper, i.e., also for $\mathbf{R}_{s,n}$ and $\mathbf{R}_m$.

V. Adaptive Volterra Filters

As a comparison, we want to show the effects of our newly derived equalizers (12) and (22) w.r.t. an adaptive equalizer solution. For that purpose we use an extension to the conventional linear LMS adaptive algorithm [25]. These nonlinear extensions are needed to adapt the second-order kernel of the nonlinear LMS equalizer during training [26]. The modified LMS updating equations are obtained as

$$\mathbf{g}_1[k] = \mathbf{g}_1[k-1] + \mu_1\mathbf{y}[k]e[k] \quad (23)$$

for the first-order kernel and as

$$\mathbf{g}_2[k] = \mathbf{g}_2[k-1] + \mu_2(\mathbf{y}[k] \boxtimes \mathbf{y}[k])e[k] \quad (24)$$

for the second-order kernel, with $e[k]$ denoting the error signal which is given as $e[k] = d[k-\delta] - \hat{d}[k]$ during the training phase.

VI. Simulation Results

In this section, we compare the linear (12) and nonlinear (22) equalizer. For that purpose we have used the nonlinear second-order equivalent system model derived in [6], which is a model for a frame-differential (FD) transmitted-reference (TR) UWB autocorrelation receiver (AcR) front end. It describes the intersymbol interference due to a time-dispersive multipath channel for the nonlinear AcR. The authors also derive an exact expression for the noise variance at the output of the receiver, given a certain double-sided noise spectral density $N_0/2$ at the receiver input. As a first approximation (which turned out to be a reasonable assumption for many scenarios), the noise is assumed to be a zero-mean i.i.d. Gaussian random process with a fixed variance σ_n^2 depending on the receiver parameters. Furthermore, the FD-TR-UWB scheme uses BPSK signaling to transmit data and thus fits into our framework defined in (4).

To compute the equalizer coefficients according to (12) and (22) it is necessary to determine the autocorrelation matrix of the data vectors $\mathbf{s}[k]$ and $\mathbf{r}[k]$. The autocorrelation matrix for $\mathbf{s}[k]$ is shown in (13) and is a scaled identity matrix with dimension $2L^2 - L$ if we assume that L is the number of channel taps and equalizer taps (i.e., $L_e = L$). The construction of $\mathbf{r}[k]$ is more complex and if we again take the same length for the channel and the equalizer, we get a huge autocorrelation matrix $\mathbf{R}_r$, of dimension $(4L^4 - 4L^3 + 3L^2 - L)/2$. We see that this size is mostly determined by the fourth-order term and brings an enormous increase in the size of the autocorrelation matrix.

To achieve reasonable simulation times and matrix sizes, a simplification of the nonlinear equivalent system model coefficients has been investigated first. For an RMS delay spread of the channel impulse response of $\tau_{\mathrm{rms}} = 10$ ns the authors in [6] propose an equivalent nonlinear system model with 17 linear coefficients and $(17 \times 16)/2$ second-order kernel coefficients at a data rate of 125 Mb/s (i.e., a symbol time of 8 ns) to achieve sufficient model accuracy. For estimating the huge correlation matrix $\mathbf{R}_r$ of the data when setting the equalizer length equal to the channel length, this complexity has to be reduced. For that reason, a comparison of the equivalent system model to a truncated version of itself has been done first. The quantity we've compared to achieve similar system behavior is the data averaged bit error rate (BER). Moreover, the quantiles of the two results are compared and analyzed by simulation. It has been seen already in [6] that the RMS value of the equivalent nonlinear system model coefficients is rather low when deviating from the desired data symbol. A sophisticated analysis of this behavior is also found in [27]. This means that the overall contribution of the terms with high distance to the desired symbol is small or even negligible.

For an equivalent truncated system model of six taps (i.e., six taps for the linear kernel and $(6 \times 5)/2$ taps for the second-order nonlinear kernel), the performance results in terms of BER are shown in Fig. 1. With the obtained results, a conventional threshold detector has been used, which decides on the sign of the sampled output signal without equalization. It is seen, that the results for the 90% quantile, mean, and median of the data-averaged BER are practically the same for the truncated system. For the 10% quantile a minor deviation from the full system model is visible. For that reason, a truncated version of the nonlinear equivalent system model is used to keep the complexity low and allow estimation of the autocorrelation matrix of the data. If we consider, that the perfect equalizer (ZF) would have an infinite number of taps (IIR) the truncation to $L_e = L = 6$ taps is rather crude. An increase to 12 taps for the equalizer length L_e has shown to deliver reasonable results. With the specified lengths of channel and equalizer the autocorrelation matrix was constructed according to Appendix II, and then used for all the computations since it remains constant.

To benchmark the performance, an adaptive nonlinear filter has been used with (23) and (24) as update equations. The length of the adaptive filter was also set to $L_e = 12$ to have similar computational cost. For the step size of the algorithm, we used constant and exponentially decaying step sizes. With a constant step size the adaptive algorithm has problems to achieve a high performant solution for the inverse of the system. With an exponentially decaying step size a high performance is observed, which is very similar to the performance of our proposed nonlinear equalizer. The comparison of the analytical equalizer computations is shown in Fig. 2 where, again, the mean, median, 10%, and 90% quantiles are shown for the achieved equalizer performance. It is clearly visible that the nonlinear equalizer

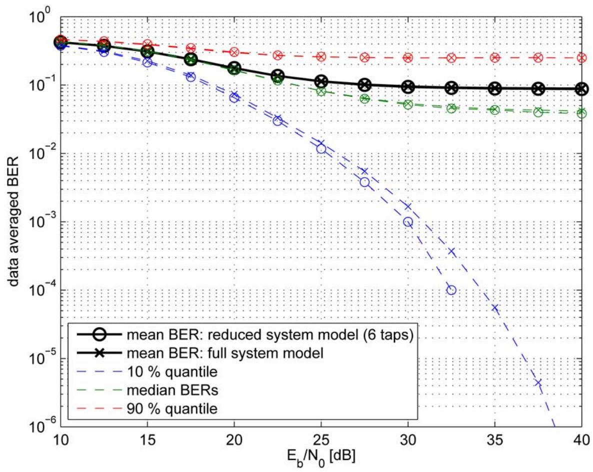

Fig. 1. Comparison of reduced equivalent system model performance with full system model performance; determined (averaged) over 1000 different equivalent system models.

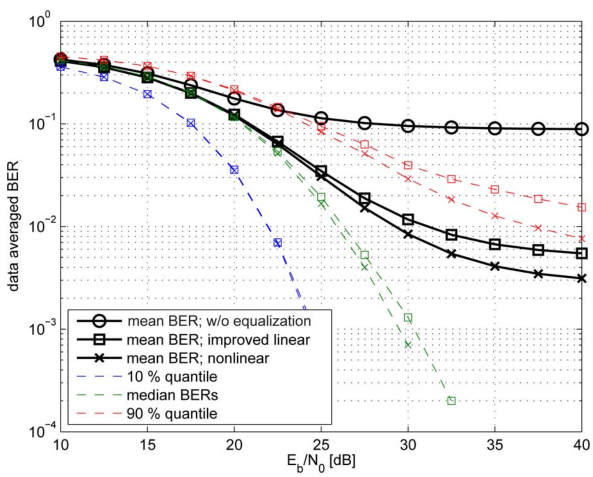

Fig. 2. Comparison of derived MMSE equalizers; linear and second-order equalizer performance analyzed over 800 different equivalent system models.

outperforms the linear equalizer. This comes, however, at the cost of an increased complexity.

The results of the adaptive equalizer are shown in Fig. 3. It is seen clearly, that a similar performance can be achieved by these equalizers. However, the estimation of the equalizer coefficients has to be performed with training sequences. For our equivalent nonlinear system model, a few tens of thousands of training symbols are needed to achieve good convergence of the coefficients of the nonlinear adaptive equalizer. A good convergence point is achieved when using an exponentially decaying step-size parameter in (24). With this decaying step-size parameter also the convergence speed is influenced which is also one of the reasons why we need a lot of training to find a good nonlinear equalizer.

We furthermore compared the output of the receiver front-end without equalization, with the improved linear equalizer, and with the nonlinear equalizer. The sequences of data symbols is depicted in Fig. 4 for an SNR of 40 dB to keep possible mistakes due to noise small. To allow a comparison to the originally transmitted data sequence we have depicted the data symbols

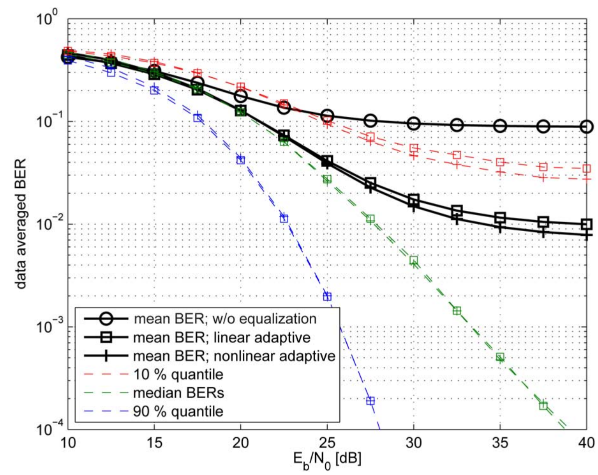

Fig. 3. Comparison of adaptive equalizers with 50 000 training symbols and an exponentially decreasing step size; Performance evaluated by analyzing over 800 equivalent system models.

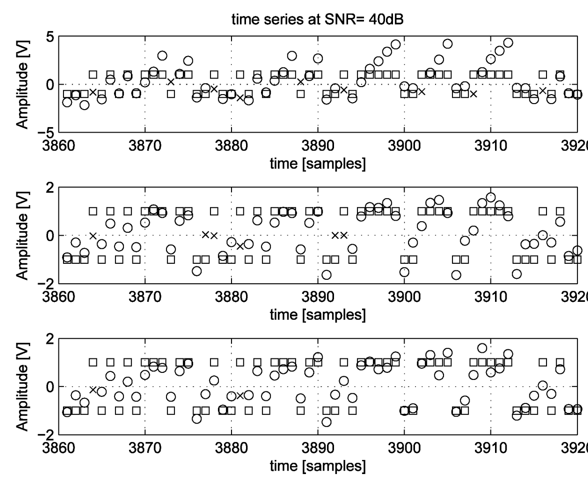

Fig. 4. Time series of distorted data symbols (top), data symbols after the linear equalizer (middle), and data symbols after nonlinear equalizer (bottom). "□" denotes the originally transmitted symbol, "o" denotes correctly detected symbol for each equalizer, and "×" denotes incorrectly detected data symbol for each equalizer, respectively.

with squares in each subplot. Furthermore, the equalized data symbols are shown either with a "×" for an incorrectly detected data symbol or a "o" for a correctly detected symbol. One can see that especially at data symbol changes (from +1 to -1, and vice versa) the nonlinear equalizer achieves better performance due to its nonlinear dynamics.

VII. Conclusion

In this paper, we have shown an equalization approach for nonlinear second-order Volterra systems. First a novel linear equalizer has been derived, which considers a nonlinear Volterra structure of the channel. The achieved results for this computation are similar to the results in [17], but are formulated more generally. Furthermore, a novel second-order Volterra equalizer was designed by explicitly solving the MMSE problem for the tandem connection of two second-order Volterra systems. Compared to the pth-order inverse shown in [16], the noise in the system has been considered in our approach. Furthermore, the

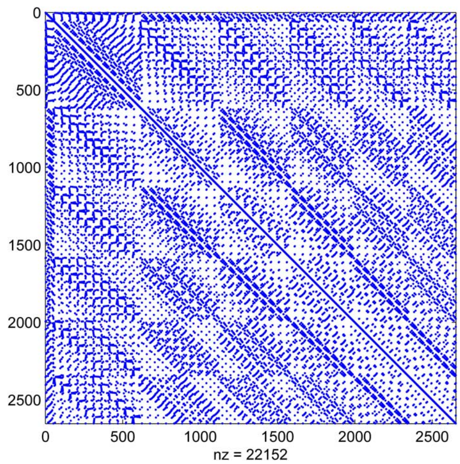

Fig. 5. Autocorrelation matrix of the data vector $\mathbf{r}[k]$. The matrix has 2652×2652 elements, where 22 152 are nonzero which reduces computational complexity because of the sparsity.

linear term is not required to be minimum phase, i.e., to be invertible, in our approach.

The performance of the novel equalizer is compared to a similarly structured adaptive equalizer, which shows that similar performances can be achieved with both approaches. However, the optimal solution is found with our expressions in one computational step when the coefficients of the nonlinear model are available. An adaptive algorithm with exponentially decaying step size requires prohibitively long training sequences to approach that optimum.

Appendix I
Commutation of the Kronecker Product

The Kronecker product of vectors is not commutative, which means that if the operands are exchanged a different result is obtained. However, it is always possible to find a permutation matrix $\mathbf{P}$ that achieves

$$\mathbf{c} = \mathbf{a} \otimes \mathbf{b} \equiv \mathbf{P}(\mathbf{b} \otimes \mathbf{a}). \tag{25}$$

If we assume that $\mathbf{a}$ is an $m \times 1$ vector and $\mathbf{b}$ is an $n \times 1$ vector, $\mathbf{c}$ is an $mn \times 1$ vector. Thus, a valid permutation matrix has to be an $mn \times mn$ matrix. If we assume that $\mathbf{E}_{xy}$ is a matrix with dimensions $m \times n$ filled up with zeros, except for a single 1 at the position (x, y) we can express the permutation matrix as

$$\mathbf{P} = [\mathrm{vec}(\mathbf{E}_{11}), \mathrm{vec}(\mathbf{E}_{12}), \ldots, \mathrm{vec}(\mathbf{E}_{mn})] \tag{26}$$

where $\mathrm{vec}(\cdot)$ is denoting the vector operator, i.e., the operator that stacks a matrix into a vector columnwise [28].

Appendix II
Correlation Matrix of the Data Terms

In this section we derive an analytical description for the correlation matrix $\mathbf{R}_r$. Since it contains all different correlations between data terms up to eighth order we can give a set of rules how to create this matrix in a systematic way. First of all, we describe the correlation matrix a bit more explicitly. The correlation between the data terms is given as $\mathsf{E}\{\mathbf{r}[k]\mathbf{r}^{\mathsf{T}}[k]\}$ where each of the data terms is given as $\mathbf{r}[k] = [\mathbf{s}^{\mathsf{T}}[k], (\mathbf{s}[k] \otimes \mathbf{s}[k])^{\mathsf{T}}]^{\mathsf{T}}$ and $\mathbf{s}[k]$ can be expressed as $\mathbf{s}[k] = [\mathbf{d}_x^{\mathsf{T}}[k], (\mathbf{d}_x[k] \boxtimes \mathbf{d}_x[k])^{\mathsf{T}}]^{\mathsf{T}}$. Substituting this, we obtain

$$\mathbf{r}[k] = \Big[\mathbf{d}_x^{\mathsf{T}}[k], (\mathbf{d}_x[k] \boxtimes \mathbf{d}_x[k])^{\mathsf{T}}, \big[\mathbf{d}_x^{\mathsf{T}}[k], (\mathbf{d}_x[k] \boxtimes \mathbf{d}_x[k])^{\mathsf{T}}\big] \otimes \big[\mathbf{d}_x^{\mathsf{T}}[k], (\mathbf{d}_x[k] \boxtimes \mathbf{d}_x[k])^{\mathsf{T}}\big]\Big]^{\mathsf{T}}. \tag{27}$$

To determine the autocorrelation matrix, the expectation operator of the outer product of (27) with itself has to be computed. If one studies the structure of the data products in detail, one can see that the matrix consists of sections with products of different orders. To analyze these products stepwise, we separate the correlation matrix in four parts, like

$$\mathbf{R}_r = \left[\begin{array}{c:c} \mathbf{R}_1 & \mathbf{R}_2 \\ \hdashline \mathbf{R}_2^{\mathsf{T}} & \mathbf{R}_3 \end{array}\right] \tag{28}$$

where the separating line between the parts is drawn right after the fourth-order parts. Since the data vector up to the fourth-order products is defined with the reduced Kronecker notation, $\mathbf{R}_1$ is exactly the same as $\mathbf{R}_s$ used for the linear equalizer. For the higher-order terms, contained in $\mathbf{R}_2$ and $\mathbf{R}_3$, we can define conditions which have to be computed for each element and then it can be decided whether there is correlation between the terms or not. For the following, we assume that all the processes are stationary. Thus we drop the time index k of the data vectors. Furthermore, in each element of $\mathbf{R}_2$ and $\mathbf{R}_3$, we get a certain amount of data symbols contributing. Under assumption A1), we can say, that for each element in the correlation matrix where an odd number of data symbols is contributing, the resulting correlation is zero. For an even number of contributing symbols, we can define conditions which have to be fulfilled such that there is correlation between the data symbols, otherwise also these data symbols are uncorrelated.

Generally, the amount of different conditions which have to be checked is huge in this case. However, this problem is very similar with a set partitioning problem in combinatorics [29]. There, Bell has specified a number (i.e., the Bell number) which gives the number of partitioned sets of one dataset. For our problem, not all different subsets are interesting. We just want to focus on the subsets which are contributing something different from zero to the autocorrelation matrix of the data vector.

For the correlation of two data symbols, we have exactly one case where this condition is fulfilled. There the contribution in the autocorrelation matrix is $\mathsf{E}\{d_i d_j\} = c^2$ for $i = j$, where d_i is the ith element of $\mathbf{d}_x[k]$. If we consider four symbols, we get already four matches which contribute a different value than zero to the correlation matrix. These four cases are split in three

$$
\begin{aligned}
\mathsf{E}\{d_i d_j\} &= c^2 \quad \text{if} \quad i=j && 1\\
\mathsf{E}\{d_i d_j d_k d_l\} &= c^4 \begin{cases} \text{if} & i=j \wedge k=l\\ & \vdots\\ \text{if} & i=j=k=l \end{cases} && 4\\
\mathsf{E}\{d_i d_j d_k d_l d_m d_n\} &= c^6 \begin{cases} \text{if} & i=j \wedge k=l \wedge m=n\\ \vdots & \\ \text{if} & i=j=k=l \wedge m=n\\ \vdots & \\ \text{if} & i=j=k=l=m=n \end{cases} && 31\\
\mathsf{E}\{d_i d_j d_k d_l d_m d_n d_o d_p\} &= c^8 \begin{cases} \text{if} & i=j \wedge k=l \wedge m=n \wedge o=p\\ \vdots & \\ \text{if} & i=j=k=l \wedge m=n \wedge o=p\\ \vdots & \\ \text{if} & i=j=k=l=m=n \wedge o=p\\ \vdots & \\ \text{if} & i=j=k=l \wedge m=n=o=p\\ \vdots & \\ \text{if} & i=j=k=l=m=n=o=p \end{cases} && 374
\end{aligned} \tag{29}
$$

sets of two data symbols and one set of four data symbols. Considering six data symbols, generally gives 31 different possible conditions. These 31 split to 15 where we have three pairs of data symbols, 15 where we have one pair of data symbols and one quad of data symbols, and one single six-element entry. For the eight-data-symbol case, we have generally 374 subsets of the 4140 (the Bell number B_8) that contribute. There we find 103 terms consisting of four pairs, 210 terms consisting of a group of four and two pairs, 25 terms consisting of a pair and a group of six elements, 35 terms consisting of two groups of four and one single entry that consists of a group of eight. This in total gives 374 different conditions, and by superimposing the other contributions with less than eight elements, we get a total number of 410 different possibilities to get a contribution in the autocorrelation matrix that is different from zero. To clarify the grouping, we depict the conditions once again in (29) where the different permutations are obtained when shifting indices.

Similar considerations are possible for the two other correlation matrices $\mathbf{R}_{s,n}$ and $\mathbf{R}_m$ and similar conditions can be formulated for the correlation terms of noise and data samples, i.e., the entries in the correlation matrix. See equation (29) at the top of the next page. To show the complicated structure of such a correlation matrix, we have generated an example for the case where channel and equalizer have the same length of six taps. One can observe some regularity in this matrix. Each element containing a value different from zero is visualized with a dot. The matrix has $2652 \times 2652 = 7033104$ entries. However, only 22 152 of them are nonzero, allowing sparse matrix computations reducing computational complexity.

REFERENCES

[1] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York: MacMillan, 1994.

[2] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*. New York: Krieger, 1980.

[3] V. J. Mathews and G. L. Sicuranza, *Polynomial Signal Processing*. New York: Wiley-Interscience, 2000.

[4] R. Hermann, "Volterra modeling of digital magnetic saturation recording channels," *IEEE Trans. Magn.*, vol. 26, pp. 2125–2127, Sep. 1990.

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

[6] K. Witrisal, G. Leus, M. Pausini, and C. Krall, "Equivalent system model and equalization of a differential impulse radio UWB system," *IEEE J. Sel. Areas Commun.*, vol. 23, pp. 1851–1862, Sep. 2005.

[7] H. Koeppl and P. Singerl, "An efficient scheme for nonlinear modeling and predistortion in mixed-signal systems," *IEEE Trans. Circuits Syst. II*, vol. 53, pp. 1368–1372, Dec. 2006.

[8] F. Gao and W. Snelgrove, "Adaptive linearization of a loudspeaker," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, Apr. 1991, vol. 5, pp. 3589–3592.

[9] E. Aschbacher and M. Rupp, "Modelling and identification of a nonlinear power-amplifier with memory for nonlinear digital adaptive predistortion," in *Proc. 4th IEEE Workshop Signal Processing Advances Wireless Communications (SPAWC)*, Jun. 15–18, 2003, pp. 658–662.

[10] J. R. Barry, E. A. Lee, and D. G. Messerschmitt, *Digital Communication*, 3rd ed. Boston, MA: Kluwer Academic, 2003.

[11] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1999.

[12] *Signal Processing for Mobile Communications Handbook*, M. Ibnkahla, Ed. Boca Raton, FL: CRC Press, 2005.

[13] J. G. Proakis, *Digital Communications*, 4th ed. New York: McGraw-Hill Int., 2001.

[14] A. Carini, J. V. Mathews, and G. L. Sicuranza, "Exact and pth order equalization and linearization of recursive polynomial systems," in *Proc. 32nd Asilomar Conf. Signals, Systems, Computers*, Pacific Grove, CA, Nov. 1998, pp. 688–692.

[15] E. Mumolo and A. Carini, "On the Stability of Discrete Time Recursive Volterra Filters," *IEEE Signal Process. Lett.*, vol. 6, pp. 230–232, Sep. 1999.

[16] M. Schetzen, "Theory of pth-order inverses of nonlinear systems," *IEEE Trans. Circuits Syst.*, vol. CAS-23, no. 5, pp. 285–291, May 1976.

[17] C. Krall, K. Witrisal, H. Koeppl, G. Leus, and M. Pausini, "Nonlinear equalization for frame-differential IR-UWB receivers," in *Proc. IEEE Conf. Ultra Wideband (ICU)*, Sep. 2005, pp. 576–581.

[18] G. B. Giannakis and E. Serpedin, "Linear multichannel blind equalizers of nonlinear FIR Volterra Channels," *IEEE Trans. Signal Process.*, vol. 45, no. 1, pp. 67–81, Jan. 1997.

[19] J. Romme and K. Witrisal, "Transmitted-reference UWB systems using weighted autocorrelation receivers," *IEEE Trans. Microw. Theory Tech.*, vol. 54, pp. 1754–1761, Apr. 2005.

[20] A. Redfern and G. Zhou, "A root method for Volterra system equalization," *IEEE Signal Process. Lett.*, vol. 5, pp. 285–288, Nov. 1998.

[21] R. Nowak and B. V. Veen, "Volterra filter equalization: A fixed point approach," *IEEE Trans. Signal Process.*, vol. 45, no. 2, pp. 377–388, Feb. 1997.

[22] S. Im, "Adaptive equalization of nonlinear digital satellite channels using a frequency-domain Volterra filter," in *Proc. Conf. IEEE Military Communications Conf. (MILCOM)*, Oct. 1996, vol. 3, pp. 843–848.

[23] G. Raz and B. V. Veen, "Blind equalization and identification of nonlinear and IIR systems-a least squares approach," *IEEE Trans. Signal Process.*, vol. 48, no. 1, pp. 192–200, Jan. 2000.

[24] V. S. Kafka and U. Appelt, "Nonlinear equalization by recursive polynomial systems with a nonlinear predictor," presented at the IEEE/EURASIP Workshop on Nonlinear Signal and Image Processing, (NSIP), Baltimore, MD, 2001.

[25] S. Haykin, *Adaptive Filter Theory*, 4th ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[26] V. J. Mathews, "Adaptive polynomial filters," *IEEE Signal Process. Mag.*, vol. 8, pp. 10–26, Jul. 1991.

[27] J. Romme, "UWB channel fading statistics and transmit-reference communication," Ph.D. dissertation, Graz Univ. of Technology, Graz, Austria, 2008.

[28] A. Graham, *Kronecker Products and Matrix Calculus With Applications*. New York: Ellis Horwood, Ltd., 1981.

[29] J. H. van Lint and R. M. Wilson, *A Course in Combinatorics*, 1st ed. Cambridge, U.K.: Cambridge Univ. Press, 1993.

Christoph Krall (M'08) was born in St. Veit/Glan, Austria, in 1979. He received the Dipl.-Ing. degree and the Ph.D. degree (with distinction) in electrical engineering from the Graz University of Technology, Austria, in 2005, and 2008, respectively.

Currently, he is a Research Engineer in the Radiocommunications Systems Division at Rohde & Schwarz GmbH & Co. KG, Munich, Germany. His research interests include wireless broadband communication systems, nonlinear and adaptive signal processing, and wireless transceiver design.

Klaus Witrisal (M'03) was born in Graz, Austria in 1972. He received the Dipl.-Ing. degree in electrical engineering from the Graz University of Technology, Austria, in 1997 and the Ph.D. degree (*cum laude*) from the Delft University of Technology, Delft, The Netherlands, in 2002.

From 1997 to 2001, he was a Research Engineer at the Delft University of Technology. From January to October 2002, he was a Project Leader with AVL, Graz, Austria. Since November 2002, he has been an Assistant Professor at the Signal Processing and Speech Communication Laboratory (SPSC) of the Graz University of Technology, where he runs a number of research projects in the area of ultra-wideband (UWB) communications. His research interests are in signal processing for broadband and UWB wireless communications, propagation channel modeling, and positioning.

Dr. Witrisal is Co-Chair of the IEEE Microwave Theory and Techniques/ Communications (MTT/COM) Society of the IEEE Austria Section.

Geert Leus (SM'05) was born in Leuven, Belgium, in 1973. He received the Electrical Engineering degree and the Ph.D. degree in applied sciences from the Katholieke Universiteit Leuven, Belgium, in 1996 and 2000, respectively.

He has been a Research Assistant and a Postdoctoral Fellow of the Fund for Scientific Research—Flanders, Belgium, from October 1996 until September 2003. During that period, he was affiliated with the Electrical Engineering Department of the Katholieke Universiteit Leuven, Belgium. Currently, he is an Associate Professor at the Faculty of Electrical Engineering, Mathematics and Computer Science of the Delft University of Technology, Delft, The Netherlands. During summer 1998, he visited Stanford University, Stanford, CA, and from March 2001 to May 2002, he was a Visiting Researcher and Lecturer at the University of Minnesota, Minneapolis. His research interests are in the area of signal processing for communications.

Dr. Leus received a 2002 IEEE Signal Processing Society Young Author Best Paper Award and a 2005 IEEE Signal Processing Society Best Paper Award. He is the Vice-Chair of the IEEE Signal Processing for Communications Technical Committee, and an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and the *EURASIP Journal on Applied Signal Processing*. In the past, he has served on the Editorial Board of the IEEE SIGNAL PROCESSING LETTERS and the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.

Heinz Koeppl received the M.Sc. degree in physics (*summa cum laude*) from Karl-Franzens University, Graz, Austria, and the Ph.D. degree in electrical engineering (*summa cum laude*) from the Graz University of Technology, Graz, Austria, in 2000 and 2004, respectively.

During 2005 and 2006, he was an Erwin Schrödinger Postdoctoral Fellow with the University of California, Berkeley, working together with Prof. L. Chua at the Department of Electrical Engineering and Computer Sciences. Subsequently, he joined the Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland, where he is currently a Postdoctoral Fellow at the Laboratory of Nonlinear Systems in the School of Communication and Computer Sciences. In 2008, he was a Visiting Research Fellow with the Systems Biology Department at Harvard Medical School, Boston, MA. His research interests include computational systems biology and modeling of nonlinear circuits.

Dr. Koeppl is recipient of the 2006 Fred-Margulies Award from the International Federation of Automatic Control (IFAC) for the best Ph.D. thesis.