Volterra Models for Digital PWM and Their Inverses

Fernando Chierchie and Sven Ole Aase

***Abstract*—In an all-digital class-D audio amplifier each signal sample is mapped into a pulse using digital pulse-width-modulation (PWM), and this intrinsically generates nonlinear distortion. This article develops discrete-time Volterra models for digital PWM. The analysis considers two types of demodulation filter, an ideal filter which provides insight into the behavior of the PWM itself, and an analog low-order demodulator filter which models the interaction of the PWM mapping with a real demodulator filter. Symmetric, trailing edge, and leading edge PWM are considered. Using the Volterra models, formulae for the inverse systems are developed. This facilitates digital precompensation of the above mentioned distortion by digitally pre-filtering the modulating signal prior to the PWM mapping. This method is first simulated and compared to pseudo-natural PWM, a well-known method for PWM distortion reduction. Moreover, the method is verified by real measurements from a physical realization of the prefilter setup using real-time digital signal processing.**

***Index Terms*—Nonlinear distortion, power amplifiers, pulse-width modulation.**

I. Introduction

PULSE-WIDTH-MODULATION (PWM) is an old modulation technique where the signal strength (or amplitude) is converted into pulse width. One of the obvious purposes for this conversion is in power heating, where the power of an electric oven is turned on and off in a controlled manner to obtain the desired temperature. Another application where PWM was introduced was in switched power converters and power supplies. In 1975 the first commercial integrated circuit for PWM control of power supplies started the development of a new industry [1].

The idea of using PWM for signal amplification, in a so-called "class-D amplifier." is also old [2], [3]. The main advantage of class-D amplifiers compared to other types of amplifiers is their high efficiency which can be above 90% in practical applications (theoretically 100% efficient), this compared to standard class-A and class-B amplifiers with 50% and 78.5% theoretical maximum efficiency, respectively, or class-AB which is a trade-off between the linearity of class-A amplifiers and the efficiency of class-B, and has a theoretical maximum efficiency below 78.5% [4].

More recent works are focused on the development of fully digital, low-distortion class-D amplifiers [5]. The modulation for class-D amplifiers can be implemented in two ways: analog or digital. In analog modulation, the (analog) modulating signal is compared to a periodic saw-tooth or triangular signal, and the output of the comparator is the PWM signal. In digital PWM, the modulating signal is a discrete-time signal, and each signal sample is mapped into the corresponding PWM pulse. The advantage of digital PWM is that the signal chain is kept digital all the way to the output stage, the disadvantage is that the direct mapping of sample height to pulse width introduces nonlinear distortion in the audio frequency band [6].

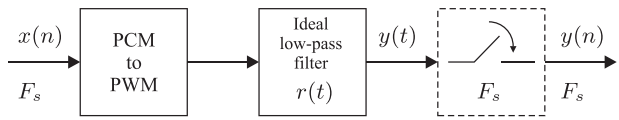

Fig. 1. Analysis of digital PWM in the discrete-time domain.

A great research effort has been centered on analyzing PWM nonlinearity in the frequency domain [6]–[8]. Also, many compensation methods, mostly based on imitating the behavior of analog PWM have been presented in recent decades, such as the pioneer work of Goldberg and Sandler [9] and also in recent works [10]. All these methods require without exception a high PWM frequency which increases power losses.

New approaches [11]–[15] have been based on the analysis of the distortion generated by digital PWM in the *discrete-time domain*, as shown in Fig. 1: The modulating signal used to generate the PWM signal is a discrete-time signal $0 \leq x(n) \leq 1$ with sampling frequency $F_s = 1/T_s$. Each sample $x(n)$ is mapped into a PWM pulse, and the resulting pulse train is passed through an ideal (brick-wall) analog filter, with cut-off frequency $F_s/2$, for reconstruction of the analog waveform $y(t)$. Since $y(t)$ is bandlimited to $F_s/2$, this signal can be represented by its samples $y(n)$ taken at frequency F_s. Note that the sampling box in Fig. 1 is printed in dotted line, thus implying that *it is not realized physically*, but merely serves as a mathematical tool to obtain a discrete-time representation of the reconstructed waveform $y(t)$. The whole idea is that with the setup presented, we can evaluate the distortion generated by digital PWM by direct comparison between the modulating signal $x(n)$ and the output $y(n)$. This was exploited in [12], [14] by modeling the relation between $x(n)$ and $y(n)$ as a Volterra filter [16]. This opened up the possibility to *precompensate* the nonlinear distortion induced by digital PWM by prefiltering the modulating signal $x(n)$ with a Volterra filter having the inverse characteristics of the model filter, thus eliminating the nonlinear distortion from the reconstructed waveform $y(t)$.

A. Contributions of This Work

The present work is both a unification of the PWM models and inverses presented in [12], [14], as well as considerable extensions: ***a***) Whereas [12], [14] only treats symmetric PWM, both leading-, trailing-, and symmetric PWM are covered, assuming either ideal-, or low-order reconstruction filter. ***b***) General model inverses are derived in Section III: Both leading-,

Manuscript received February 03, 2015; revised May 13, 2015 June 24, 2015 and August 13, 2015; accepted August 17, 2015. Date of current version September 25, 2015. F. Chierchie would like to acknowledge the support of the BEC.AR program of the Jefatura de Gabinete de Ministros de la Nación. This paper was recommended by Associate Editor H. Johansson.

F. Chierchie is with the Instituto de Investigaciones en Ing. Eléctrica (IIIE) Alfredo Desages (UNS-CONICET). Depto. de Ing. Eléctrica y de Computadoras. Universidad Nacional del Sur, 8000 Bahía Blanca, Argentina (e-mail: fernando.chierchie@uns.edu.ar).

S. O. Aase is with the Department of Electrical Engineering and Computer Science, the University of Stavanger, 4036 Stavanger, Norway (e-mail: sven.o.aase@uis.no).

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/TCSI.2015.2476299

trailing-, and symmetric PWM are covered, assuming either ideal-, or low-order reconstruction filter. ***c***) For the ideal reconstruction filter, the work in [12] is extended by deriving explicit, closed-form expressions for the Volterra kernels. ***d***) In [12], [14], the performance of the prefilter was simulated using MATLAB, assuming symmetric PWM. The present work demonstrates the utility of the Volterra prefilter method using two different experimental setups:

i) A comparison with an established method under tough working conditions: Simulink is used to test the prefilter's ability to remove strong nonlinearities due to very low sampling (and PWM) frequency (88.2 kHz and 176.4 kHz). The performance is favorably compared to a well-known algorithm for pseudo-natural PWM (PNPWM) [5], [6], [17].

ii) Real measurements from a physical realization of the prefilter using real-time DSP: Here we demonstrate the success of the prefilter in a setup handling practical issues such as finite pulse width precision, which is dealt with using noise feedback coding (NFC). Here a higher switching frequency (176 kHz) is used (still very low compared to standard class-D applications), allowing lower prefilter order and better working conditions for the NFC module.

The article is organized as follows: In Section II we derive the Volterra models for different types of PWM symmetries and using different types of demodulator low-pass filters. In Section III inverse models that facilitate precompensation for the nonlinear distortion induced by digital PWM are presented. Simulations, comparisons and experiments are presented in Section IV and conclusions are elaborated in Section V.

II. Volterra Models for Digital PWM

The three most common types of PWM are: 1) Symmetric PWM, where the pulse is centered on a fixed point and both edges of the pulses are modulated; 2) Trailing edge PWM, where the raising edge is fixed and the falling edge is modulated; 3) Leading edge PWM where the falling edge is fixed and the raising edge is modulated. In this section we analyze each type of PWM, first assuming an ideal, analog demodulation filter as shown in Fig. 1, then assuming a physically implementable low-order analog filter and a more elaborate setup to be introduced later. It will be shown that for all cases, the model describing the input-output relation is a diagonal Volterra filter. We therefore start with a short introduction to Volterra filters.

A. Definition of the General and Diagonal Volterra Filter

If $x(n)$ is the input to a Volterra filter then the output $y(n)$ can be written as

$$y(n) = \sum_{q=1}^{\infty} \gamma_q(n)$$

where

$$\gamma_q(n) = \sum_{i_1} \cdots \sum_{i_q} h_q(i_1, \ldots, i_q) x(n-i_1) \cdots x(n-i_q), \quad (1)$$

where $h_q(i_1, \ldots, i_q)$ can be considered as a qth order impulse response (kernel) characterizing the nonlinear behavior of the system [18].

A simpler system is obtained if only the kernel values for identical indexes $i_1 = \cdots = i_q = i$ are non-zero: $h_q(i, \ldots, i) \equiv h_q(i) \neq 0$. In this case, the general equation (1) for the nonlinear Volterra system can be simplified to

$$\gamma_q(n) = \sum_i h_q(i) x(n-i)^q = h_q(n) * x(n)^q,$$

where "$*$" denotes discrete-time convolution. This gives as output

$$y(n) = \sum_{q=1}^{\infty} h_q(n) * x(n)^q, \quad (2)$$

which is a diagonal Volterra model, or sometimes denoted a parallel Hammerstein structure [16].

B. General Setup and Assumptions

The modulating analog signal $x(t)$, where $0 \leq x(t) \leq 1$ has a maximum frequency component F_x, where $F_x < F_s/2$. This signal is sampled at frequency F_s to obtain the discrete-time representation $x(n)$. Each sample $x(n)$ is used to construct a pulse of width $x(n)T_s$ and height 1. Full-range modulation is considered in the sense that if $x(n) = 1 \forall n$, adjacent pulses touch since the pulse width is T_s. In this case, using a passive, analog low-pass filter with unity passband gain, will result in the reconstruction of a time-continuous signal $y(t) = 1$. Also note that this system is equivalent to a zero-centered system with $x(n) \in [-1/2, 1/2]$ and pulse amplitudes $\pm 1/2$.

The general procedure to compute the Volterra kernels for the different types of PWM symmetry and different types of output filter, is based on computing the convolution integral between a single pulse $p(t)$ and the impulse response $r(t)$ of the output demodulator filer. Since $r(t)$ is a linear time-invariant system, the complete output due to a PWM signal, composed of shifted pulses of different widths, can be computed by shifting and adding the output due to each pulse. After some considerations, sampling of the finite bandwidth output signal results in nonlinear, discrete-time models for the different types of PWM.

To demodulate the PWM signal an analog low-pass filter is typically used. In this work, two kinds of low-pass filters are considered, first an ideal $\mathrm{sinc}(\cdot)$, or so-called brick-wall filter is used and then, a low-order filter is studied. Each filter leads to a specific Volterra model of PWM.

C. Volterra Kernels Using an Ideal Analog Low-Pass Filter

In this subsection we consider an ideal analog low-pass filter for the demodulation of the PWM signal. The impulse response of this filter with cutoff frequency $F_s/2$ and unity passband gain is [19]

$$r(t) = F_s \frac{\sin(\pi F_s t)}{\pi F_s t} = F_s \mathrm{sinc}(F_s t) \quad (3)$$

with $F_s = 1/T_s$.

The scheme used to find the Volterra model for PWM with an ideal low-pass demodulator filter is shown in the block diagram of Fig. 1. The modulating signal samples $x(n)$ are individually mapped into PWM pulses in the PCM to PWM block, the output is low-pass filtered using the filter in (3) to obtain the signal $y(t)$ and finally sampling at F_s gives $y(n)$.

First the output $y(t)$ in Fig. 1 due to a *single pulse* is computed for all the three types of symmetry. Then, the complete output due to a PWM signal is calculated.

Symmetric PWM and Ideal Filter: In digital PWM each signal sample is mapped into a pulse. Fig. 2(a) shows how a single input sample $x(0)$ is mapped into a single pulse $p(t)$

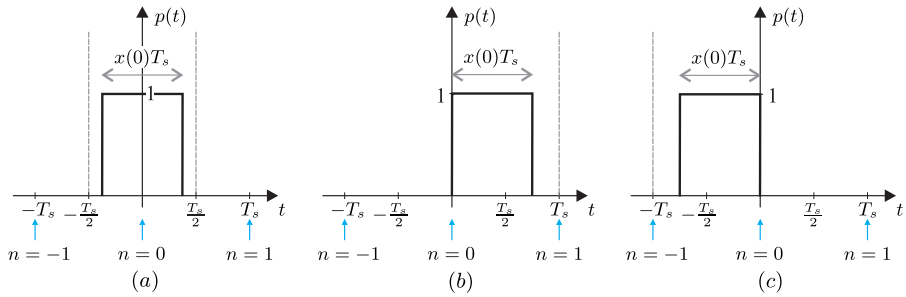

Fig. 2. Single pulse $p(t)$ used to compute the convolution integral for the ideal low-pass filter. (a) Symmetric PWM. (b) Trailing edge PWM. (c) Leading edge PWM. The up-arrows indicate the sampling instants used to compute $y_n(x)$.

centered at $t = 0$, of height one and width $x(0)T_s$. This pulse produces a response $y(t)$ at the output of the ideal filter $r(t)$ that is calculated using the convolution integral

$$\begin{aligned} y(t) &= \int_{-\infty}^{\infty} r(\tau)p(t-\tau)d\tau \\ &= \int_{t-x(0)\frac{T_s}{2}}^{t+x(0)\frac{T_s}{2}} r(\tau)d\tau \\ &= R\left(t+x(0)\frac{T_s}{2}\right) - R\left(t-x(0)\frac{T_s}{2}\right) \end{aligned}$$

where $dR(t)/dt = r(t)$.

The signal $y(t)$ can be sampled at $t = nT_s$ without aliasing problems because the filter $r(t)$ removes all frequency components above $F_s/2$. The sampling instants are indicated in Fig. 2 using arrows. Writing $x(0) = x$, the sampled output is

$$y_n(x) \equiv y(t)|_{t=nT_s} = R\left(T_s\left(n+\frac{x}{2}\right)\right) - R\left(T_s\left(n-\frac{x}{2}\right)\right) \quad (4)$$

For each n, $y_n(x)$ is a nonlinear function of x which can be expanded as a power series around zero. It is seen from (4) that $y_n(x)$ is an odd function of x: $y_n(-x) = -y_n(x)$, and this results in an odd power series expansion for each n:

$$\begin{aligned} &\vdots \\ y_{-2}(x) &= a_{-21}x + a_{-23}x^3 + a_{-25}x^5 + \dots \\ y_{-1}(x) &= a_{-11}x + a_{-13}x^3 + a_{-15}x^5 + \dots \\ y_0(x) &= a_{01}x + a_{03}x^3 + a_{05}x^5 + \dots \\ y_1(x) &= a_{11}x + a_{13}x^3 + a_{15}x^5 + \dots \\ y_2(x) &= a_{21}x + a_{23}x^3 + a_{25}x^5 + \dots \\ &\vdots \\ &\downarrow \quad\quad \downarrow \quad\quad \downarrow \\ &h_1^S(\cdot) \quad h_3^S(\cdot) \quad h_5^S(\cdot) \end{aligned} \quad (5)$$

Equation (5) also indicates how to construct the impulse responses $h_m^S(n) = a_{nm}$ for symmetric PWM (superscript S) using the power series expansion. As will be shown at the end of this section, these impulse responses are the kernels of a diagonal Volterra system as described in Section II-A. To find closed form expressions for the kernels $h_1^S(n)$, $h_3^S(n)$, $h_5^S(n)$, ... in (5), the expressions for $y_n(x)$ in both (5) and (4) can be differentiated m-times with respect to x. Then setting $x = 0$ we obtain

$$y_n^{(m)}(0) = m!h_m^S(n),$$

from (5), and

$$y_n^{(m)}(0) = \frac{T_s^m}{2^{m-1}} r^{(m-1)}(nT_s),$$

from (4), and equating the results we finally obtain

$$h_m^S(n) = \frac{T_s^m}{m!2^{m-1}} r^{(m-1)}(nT_s),\ m = 1, 3, 5, \dots \quad (6)$$

where $r^{(m)}(\cdot)$ is the m-times derivative of the impulse response of the demodulation filter given in (3).

Trailing Edge PWM and Ideal Filter: The kernels are computed following the same procedure that was used for symmetric PWM but taking into account the non-symmetric characteristics of the trailing edge pulses, as shown in Fig. 2(b). Due to this property of the pulse, the odd symmetry of (4) is lost and as a result the power series expansion of $y_n(x)$ for trailing edge also includes even powers of x. The Volterra kernels for trailing edge PWM are found as

$$h_m^{Te}(n) = T_s^m \frac{(-1)^{m-1}}{m!} r^{(m-1)}(nT_s),\ m = 1, 2, 3, \dots \quad (7)$$

In addition to the inclusion of the even powered kernels, (7) shows that compared to the symmetric PWM kernels in (6) the $1/2^{m-1}$ attenuation factor is not present for trailing edge. This means that the amplitude of the kernels for $m > 1$ are higher and it follows that trailing edge exhibits more nonlinearities than symmetric PWM. This is in agreement with previous knowledge of PWM which states that symmetric PWM has reduced distortion compared to trailing edge PWM; this has been demonstrated using frequency domain methods [6], [7].

Leading Edge PWM and Ideal Filter: For leading edge PWM the pulse in Fig. 2(c) is used to compute the convolution. Again, the even ordered Volterra kernels are present, and following the same procedure as before we get:

$$h_m^{Le}(n) = \frac{T_s^m}{m!} r^{(m-1)}(nT_s),\ m = 1, 2, 3, \dots \quad (8)$$

As for the case of trailing edge modulation the $1/2^{m-1}$ factor is not present. The difference between $h_m^{Te}(n)$ and $h_m^{Le}(n)$ is in the sign of the even kernels due to the $(-1)^{m-1}$ component.

For the three types of PWM the kernels in (6), (7), and (8) involve the computation of the derivative of the impulse response of the ideal low-pass filter. The derivative of $r(t) = \sin(\pi F_s t)/(\pi t)$ can be easily computed using the quotient rule for derivatives. Table I summarizes all the Volterra kernels of order 1 to 5 when an ideal low-pass filer is used for PWM demodulation.

It is worth noting that for the kernel of order $m = 1$ all three types of symmetry present the same result: $h_1^S(n) = h_1^{Te}(n) = h_1^{Le}(n) = \delta(n)$, a discrete-time impulse which is an identity operator. This implies that digital PWM in the forms discussed here do not induce *linear* distortion.

TABLE I
KERNELS OF VOLTERRA MODELS FOR DIGITAL PWM USING IDEAL LOW-PASS FILTER

	Type of PWM symmetry		
m	Symmetric: $h_m^S(n)$	Trailing Edge: $h_m^{Te}(n)$	Leading Edge: $h_m^{Le}(n)$
1	$\delta(n)$	$\delta(n)$	$\delta(n)$
2	0	$\begin{cases}(-1)^{n+1}/(2n) & \text{if } n\neq 0\\ 0 & \text{if } n=0\end{cases}$	$\begin{cases}(-1)^{n}/(2n) & \text{if } n\neq 0\\ 0 & \text{if } n=0\end{cases}$
3	$\begin{cases}(-1)^{n+1}/(12n^2) & \text{if } n\neq 0\\ -\pi^2/72 & \text{if } n=0\end{cases}$	$\begin{cases}(-1)^{n+1}/(3n^2) & \text{if } n\neq 0\\ -\pi^2/18 & \text{if } n=0\end{cases}$	$\begin{cases}(-1)^{n+1}/(3n^2) & \text{if } n\neq 0\\ -\pi^2/18 & \text{if } n=0\end{cases}$
4	0	$\begin{cases}(-1)^{n}\left(\pi^2n^2-6\right)/(24n^3) & \text{if } n\neq 0\\ 0 & \text{if } n=0\end{cases}$	$\begin{cases}(-1)^{n+1}\left(\pi^2n^2-6\right)/(24n^3) & \text{if } n\neq 0\\ 0 & \text{if } n=0\end{cases}$
5	$\begin{cases}(-1)^{n}\left(\pi^2n^2-6\right)/(480n^4) & \text{if } n\neq 0\\ \pi^4/9600 & \text{if } n=0\end{cases}$	$\begin{cases}(-1)^{n}\left(\pi^2n^2-6\right)/(30n^4) & \text{if } n\neq 0\\ \pi^4/600 & \text{if } n=0\end{cases}$	$\begin{cases}(-1)^{n}\left(\pi^2n^2-6\right)/(30n^4) & \text{if } n\neq 0\\ \pi^4/600 & \text{if } n=0\end{cases}$

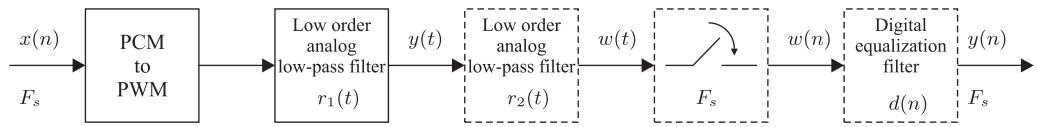

Fig. 3. Block diagram to find the PWM model when using a low order analog low-pass filter $r_1(t)$.

It should also be pointed out that for the case of trailing and leading edge modulation, the even-numbered kernels falls off as $1/n$, whereas the odd numbered kernels falls off as $1/n^2$, as in the case of symmetric modulation. This reflects the fact that trailing and leading edge modulation induce much more nonlinear distortion than the symmetric modulation.

A Complete Diagonal Volterra Model Using an Ideal Analog Filter: Since the low-pass filter is a linear time-invariant system, the complete output of the filter due to a PWM signal can be computed, using superposition, as the sum of the shifted-outputs $y_{n-i}(x(i))$ produced by each pulse of width $x(i)T_s$ constituting the PWM signal. For symmetric PWM the complete output is

$$y(n)=\sum_{i=-\infty}^{\infty} y_{n-i}\left(x(i)\right)=\sum_{q=1}^{\infty}\sum_{i=-\infty}^{\infty} h_{2q-1}^{S}(n-i)x(i)^{2q-1}$$
$$=\sum_{q=1}^{\infty} h_{2q-1}^{S}(n)*x(n)^{2q-1} \tag{9}$$

which is the sum of the outputs of the discrete-time convolution (indicated with $*$) of the kernel $h_{2q-1}^S(n)$ and the odd powers $2q-1$ of the modulating signal $x(n)$. For trailing and leading edge PWM the even kernels and even powers of the modulating signal are also present:

$$y(n)=\sum_{q=1}^{\infty} h_q(n)*x(n)^q \tag{10}$$

where the kernel $h_q(n)$ can be either $h_q^{Te}(n)$ for trailing edge or $h_q^{Le}(n)$ for leading edge. For the three types of symmetry, (9) and (10) represent a diagonal Volterra model as in (2).

D. Volterra Kernels Using a Low-Order Analog Low-Pass Filer

In this section a low-order, analog, causal low-pass filter with impulse response $r_1(t)$ is considered for the demodulation of the PWM signal. Compared to the ideal filter $r(t)$, the filter $r_1(t)$ has a finite attenuation of the higher frequency components (above $F_s/2$) of the PWM signal, because of this, some extra considerations must be made before the sampling takes place.

The block diagram considered for the development of this model, first introduced in [14], is shown in Fig. 3. The input signal $x(n)$ sampled at F_s is mapped into a PWM signal in the pulse code modulation (PCM) to PWM block. After that, the low order analog low-pass filter with impulse response $r_1(t)$ is used to filter the PWM signal. In an actual amplifier implementation the output signal $y(t)$ is the voltage feeding the loudspeaker. The objective is to find a discrete-time representation $y(n)$ of $y(t)$ in the frequency range 0 to $F_s/2$. Direct sampling of $y(t)$ would lead to aliasing, instead of this, the dashed boxes in Fig. 3 are used to find the model. Again, we emphasize that the dashed boxes are not part of a real implementation, they are only used for modeling purposes.

A low order analog filter with impulse response $r_2(t)$ is used to reduce, to a negligible level, the frequency components of $y(t)$ above $F_s/2$. After that, the signal is sampled at F_s and finally a digital equalization filter with unit pulse response $d(n)$ is used to compensate the pass-band effects of the filter $r_2(t)$. In this way the signal $y(n)$ is a discrete-time representation of the actual analog signal $y(t)$ in the 0 to $F_s/2$ frequency range.

First the output $w(t)$ in Fig. 3 due to a single pulse is computed for all the three types of symmetry. Then, the complete output due to a PWM signal is calculated, and the equalization filer $d(n)$ is included to find $y(n)$.

Symmetric PWM and Low-Order Analog Filter: In this case the location of the pulse due to the sample $x(0)$ is defined in Fig. 4(a). The convolution is computed between a single pulse of width $T_sx(0)$ and the impulse response that results from the serial connection of the two filters: $r_{12}(t)=r_1(t)*r_2(t)$. Since both filters are causal, the total filter $r_{12}(t)$ is also causal, so $r_{12}(t)=0$ for $t<0$. This situation is different from that of the ideal filter and leads to three possible regions for the convolution integral: non-overlap, partial overlap, and complete overlap. For

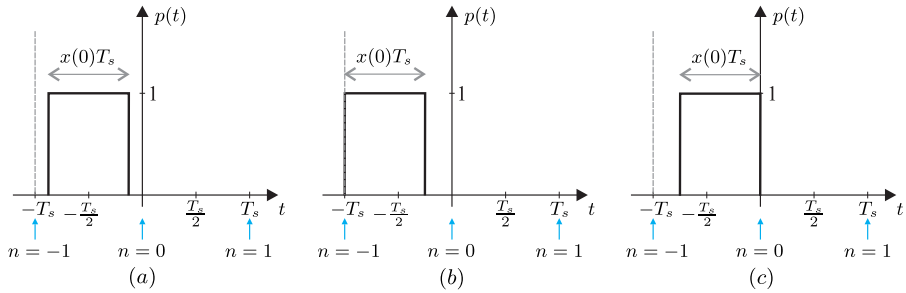

Fig. 4. Single pulse $p(t)$ used to compute the convolution integral for the causal, low-order low-pass filter. (a) Symmetric PWM. (b) Trailing edge PWM. (c) Leading edge PWM. The up-arrows indicate the sampling instants used to compute $w_n(x)$.

these three regions the output $w(t)$ of the total filter $r_{12}(t)$ can be computed as

$$w(t) = \begin{cases} 0 & \text{if } t \le -t_2 \\ \int_0^{t+t_2} r_{12}(\tau)d\tau & \text{if } -t_2 < t < -t_1 \\ \int_{t+t_1}^{t+t_2} r_{12}(\tau)d\tau & \text{if } t \ge -t_1. \end{cases}$$
$$= \begin{cases} 0 & \text{if } t \le -t_2 \\ R_{12}(t+t_2) - R_{12}(0) & \text{if } -t_2 < t < -t_1 \\ R_{12}(t+t_2) - R_{12}(t+t_1) & \text{if } t \ge -t_1. \end{cases} \tag{11}$$

where $t_1 = [1 - x(0)]T_s/2$, $t_2 = [1 + x(0)]T_s/2$ and $dR_{12}(t)/dt = r_{12}(t)$.

The signal in (11) is sampled at $t = nT_s$ at the sampling instants indicated by the arrows in Fig. 4, and writing $x(0) = x$ we have

$$w_n(x) = R_{12}\left(\left(n + \frac{1}{2} + \frac{x}{2}\right)T_s\right) - R_{12}\left(\left(n + \frac{1}{2} - \frac{x}{2}\right)T_s\right),\ n \ge 0, \tag{12}$$

and $w_n(x) = 0$ when $n < 0$. It worth noting that when sampling the signal, the partial overlap region in (11) is never sampled. This is because $-T_s \le -t_2 \le -T_s/2$ and $-T_s/2 \le -t_1 \le 0$ and hence the output $w(t)$ in the partial overlap region: $-t_2 < t < -t_1$, is always between two samples. The advantage of this carefully chosen sampling strategy is that (12) exhibit the same odd symmetry as (4), and the resulting Volterra kernels will be zero for even order.

As for the case of the ideal filter, $w_n(x)$ is a nonlinear function of x which can be expanded as a power series around $x = 0$:

$$\begin{aligned} w_0(x) &= a_{01}x + a_{03}x^3 + a_{05}x^5 + \dots \\ w_1(x) &= a_{11}x + a_{13}x^3 + a_{15}x^5 + \dots \\ w_2(x) &= a_{21}x + a_{23}x^3 + a_{25}x^5 + \dots \\ &\vdots \\ & \overset{\downarrow}{g_1^S}(\cdot) \quad \overset{\downarrow}{g_3^S}(\cdot) \quad \overset{\downarrow}{g_5^S}(\cdot) \end{aligned} \tag{13}$$

where $g_m^S(n) = a_{nm}$. The difference between (13) and the expansion for the ideal filter in (5) is that due to the causality of the analog filters, all kernels are also causal.

To find a closed form expressions for the kernels $g_m^S(n)$ in (13), the same procedure as for the ideal filter is used. Equations (12) and (13) for $w_n(x)$ are differentiated m-times with respect to x, then setting $x = 0$ and equating results, we obtain

$$g_m^S(n) = \frac{T_s^m}{m!2^{m-1}} r_{12}^{(m-1)}\left(\left(n + \frac{1}{2}\right)T_s\right),\ m = 1, 3, 5, \dots \tag{14}$$

where $r_{12}^{(m)}(\cdot)$ is the m-times derivative of the total impulse response. Compared to the ideal filter in (6), the low-order analog filter kernels $g_m^S(n)$ in (14) include half a sample shift in the sampling instant and they are causal, while $h_m^S(n)$ in (6) is not.

Trailing Edge PWM and Low-Order Analog Filter: Using the pulse in Fig. 4(b), the kernels are computed following the same procedure as above to obtain

$$g_m^{Te}(n) = T_s^m \frac{(-1)^{m-1}}{m!} r_{12}^{(m-1)}\left((n+1)T_s\right),\ m = 1, 2, 3 \dots \tag{15}$$

As for the case of the ideal filer, when compared to the symmetric case, the $1/2^{m-1}$ factor is missing showing the larger distortion of trailing edge PWM. Comparing the results in (15) with the results for trailing edge modulation and the ideal filter in (7) it can be observed that the sampling instant is shifted by one sample. This is just a consequence of the sampling scheme shown in Fig. 4(b) and results in kernels that start taking non-zero values at $n = 0$.

Leading Edge PWM and Low-Order Analog Filter: Following the same procedure as before but with the pulse in Fig. 4(c), we obtain

$$g_m^{Le}(n) = \frac{T_s^m}{m!} r_{12}^{(m-1)}(nT_s),\ m = 1, 2, 3 \cdots \tag{16}$$

As for the case of trailing edge modulation the $1/2^{m-1}$ factor is not present. Apart from the one-sample time shift, the difference between $g_m^{Te}(n)$ and $g_m^{Le}(n)$ is in the sign of the even kernels due to the $(-1)^{m-1}$ component. For leading edge and low-order analog filter the sampling instants of $r_{12}^{(m-1)}(\cdot)$ are the same as for the ideal filter kernel, $r^{(m-1)}(\cdot)$ in (8).

A Complete Diagonal Volterra Model Using a Low-Order Analog Filter: Since the filter $r_{12}(t)$ is a linear time-invariant system, the complete output of the analog filters due to a PWM signal can be computed as the sum of the shifted-outputs $w_{n-i}(x(i))$ produced by each pulse of width $x(i)T_s$ constituting the PWM signal. For symmetric PWM the output is

$$w(n) = \sum_{i=-\infty}^{n} w_{n-i}\left(x(i)\right) = \sum_{q=1}^{\infty} g_{2q-1}^S(n) * x(n)^{2q-1}. \tag{17}$$

For trailing and leading edge the even powers are also included

$$w(n) = \sum_{q=1}^{\infty} g_q(n) * x(n)^q, \tag{18}$$

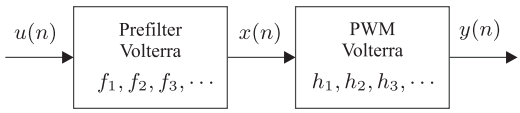

Fig. 5. Equalizing PWM using a prefilter: The audio signal $u(n)$ is processed prior to the PWM mapping.

where $g_q(n)$ can be either $g_m^{Te}(n)$ in (15), or $g_m^{Le}(n)$ in (16).

Following Fig. 3, the final step to obtain the complete kernels is to include the equalization filter $d(n)$

$$\begin{aligned} y(n) = w(n) * d(n) &= \sum_{q=1}^{\infty} d(n) * g_{2q-1}^{S}(n) * x(n)^{2q-1} \\ &= \sum_{q=1}^{\infty} h_{2q-1}^{S}(n) * x(n)^{2q-1} \end{aligned}$$

where

$$h_m^S(n) = g_m^S(n) * d(n),\ m = 1, 3, 5, \ldots,$$

are the Volterra kernels for symmetric PWM (14). Using (18), the complete kernels for trailing and leading edge PWM are: $h_m^{Te}(n) = g_m^{Te}(n) * d(n)$ and $h_m^{Le}(n) = g_m^{Le}(n) * d(n)$, respectively.

III. The Model Inverses

Having developed Volterra models for various types of digital PWM, we now focus on their inverses. Knowledge of the inverse model enables us to precompensate for the nonlinear distortion induced by digital PWM. Fig. 5 shows the procedure: Given a Volterra model characterized by the kernels $h_1, h_2, h_3, \ldots$, find the *inverse* Volterra model characterized by the kernels $f_1, f_2, f_3, \ldots$, so that the serial connection of the two systems results in the identity operator.

The problem of finding an inverse Volterra filter was addressed for general Volterra filters by Fang *et al.* [20]. Using a multidimensional z-transform methodology, they managed to construct pth order linearization operators of a nonlinear system. Following the results of [20], define the qth order nonlinear transfer function of a Volterra filter H by

$$H_q(z_1, \ldots, z_q) = \sum_{i_1} \cdots \sum_{i_q} h_q(i_1, \ldots, i_q) z_1^{-i_1} \cdots z_q^{-i_q}. \tag{19}$$

In [20] it was found that the general prefilter equalizer, or pre-inverse, was given by the following nonlinear transfer functions:

$$\begin{aligned} F_1(z_1) &= H_1(z_1)^{-1} \\ F_p(z_1, \ldots, z_p) &= -H_1(z_1 z_2 \cdots z_p)^{-1} \\ &\quad \times \sum_{m=2}^{p} \sum_{k_1+\ldots+k_m=p} H_m(z_1 \cdots z_{k_1}, \ldots, \\ &\qquad z_{p-k_m+1} \cdots z_p) \\ &\quad \times F_{k_1}(z_1, \ldots, z_{k_1}) \cdots \\ &\quad F_{k_m}(z_{p-k_m+1}, \ldots, z_p),\ p = 2, 3, \ldots, \end{aligned} \tag{20}$$

where the second summation is over all possible combinations of integers $k_1, \ldots, k_m$ so that $k_1 + \ldots + k_m = p$. For example, for $p = 2$ and $m = 2$ only the constellation $(k_1 k_2) = (11)$ is included in the sum.

All Volterra models developed in the previous section have diagonal kernel functions, and this simplifies the expressions for the inverse systems. This was shown in [12], [14], where formulae for the case of symmetric PWM/ideal filter [12] and symmetric PWM/low-order analog filter [14] were developed.

A. Preliminaries for Construction of Model Inverse

The prefilter in Fig. 5 is a Volterra filter, so the output $x(n)$ can be written as

$$x(n) = \sum_{p=1}^{\infty} x_p(n),$$

where the pth term in the sum is given as

$$x_p(n) = \sum_{i_1} \cdots \sum_{i_p} f_p(i_1, \ldots, i_p) u(n - i_1) \cdots u(n - i_p), \tag{21}$$

where $f_p(\cdot)$ is the pth order unit pulse response, or Volterra kernel function. Its p-dimensional z-transform is the pth order nonlinear transfer function $F_p(\cdot)$ defined in the same manner as in (19). In practice the Pth order approximation $\hat{x}(n) = \sum_{p=1}^{P} x_p(n)$ to the true prefilter output $x(n)$ will be used to precompensate the distortion induced by PWM.

Due to the diagonal form of the Volterra filter kernels $h_m(\cdot)$ found in Section II, the H_m term in (20) can be simplified as

$$\begin{aligned} &H_m(z_1 \cdots z_{k_1}, \ldots, z_{p-k_m+1} \cdots z_p) \\ &\quad = \sum_{i_1} \cdots \sum_{i_m} h_m(i_1, \ldots, i_m) \, (z_1 \cdots z_{k_1})^{-i_1} \\ &\qquad \cdots (z_{p-k_m+1} \cdots z_p)^{-i_m} \\ &\quad = \sum_{i} h'_m(i) (z_1 \cdots z_p)^{-i} \\ &\quad = H'_m(z_1 \cdots z_p), \end{aligned} \tag{22}$$

where $H'_m(\cdot)$ denotes the associated diagonal form.

The construction of a model inverse relies on the following 3 theorems. The first theorem only applies when using symmetric PWM, whereas Theorem 2 and 3 directly result from the diagonal property of the kernels.

Theorem 1: In the case of *symmetric* PWM, the prefilter transfer functions are zero for even indexes, that is $F_p = 0$ for $p = 2, 4, 6, \ldots$.

This theorem implies that, in (20), in the second summation we only have to include constellations where all the k_is are odd.

Theorem 2: Given m positive integers $k'_1, \ldots, k'_m$ so that $k'_1 + \ldots + k'_m = p$, denote by Q_m the number of different permutations of this set. Then

$$\sum_{k_1+\ldots+k_m=p} H'_m F_{k_1} \cdots F_{k_m} = Q_m H'_m F_{k'_1} \cdots F_{k'_m}. \tag{23}$$

In (20), the second summation is over all possible combinations of integers $k_1, \ldots, k_m$ so that $k_1 + \ldots + k_m = p$. Theorem 2 states that all combinations give identical terms in this sum.

Theorem 3: Given p, and m positive integers $k_1, \ldots, k_m$ so that $k_1 + \ldots + k_m = p$, the z-domain term

$$H'_m(z_1 \cdots z_p) F_{k_1}(z_1, \ldots, z_{k_1}) \cdots F_{k_m}(z_{p-k_m+1}, \ldots, z_p), \tag{24}$$

corresponds to the time domain expression

$$\sum_{i} h_m(i) x_{k_1}(n - i) \cdots x_{k_m}(n - i). \tag{25}$$

The general time-domain input-output relation for the pth order nonlinear transfer function F_p of the prefilter is given in (21). Even though all the model kernels $h_m(\cdot)$ are diagonal, the prefilter kernels will *not* be diagonal. However, Theorem 3

states an alternative form of implementation of the p order prefilter component where the input signals $u(n-i_1)\cdots u(n-i_p)$ are replaced by the product of previously computed terms $x_{k_1}(n)x_{k_2}(n)\cdots x_{k_m}(n)$ convoluted with the (diagonal) model kernel $h_m(\cdot)$. In (20), the pth order nonlinear transfer function F_p of the prefilter is built as a sum of components such as that given in (24), and then a final linear filter $H_1(\cdot)^{-1}$ is applied to the sum.

For proof of Theorems 1–3, see [12].

B. General, Order-Recursive, Discrete-Time Equations for Model Inverse

With the preliminaries above we are now ready to derive a general set of formulae for the inverse of all Volterra models presented in Section II. With the methodology given here, any prefilter order can be derived, but we limit the derivation to order 5. The kernel functions denoted $h_m(n)$ can be any of the previously derived $h_m^S(n)$, $h_m^{Te}(n)$, or $h_m^{Le}(n)$ either using an ideal-, or a low-order analog low pass filter. The resulting equations are order-recursive in the sense that the expression for $x_p(n)$ is dependent on $x_1(n), x_2(n), \ldots, x_{p-1}(n)$.

1) First Order: For the first order we have

$$F_1(z_1) = H_1(z_1)^{-1}, \text{ so therefore}$$
$$x_1(n) = \sum_i h_1^{-1}(i)u(n-i) = h_1^{-1}(n) * u(n), \quad (26)$$

where $h_1^{-1}(n)$ is the unit pulse response of the inverse filter $H_1(z)^{-1}$.

2) Second Order: In (20) only $m=2$ need to be considered, and here the only constellation is $(k_1k_2)=(11)$. This gives

$$F_2(z_1,z_2) = -H_1(z_1z_2)^{-1}H_2'(z_1z_2)F_1(z_1)F_1(z_2),$$

and by Theorem 3 we get

$$\begin{aligned} x_2(n) &= -h_1^{-1}(n) * \sum_i h_2(i)x_1(n-i)^2 \\ &= -h_1^{-1}(n) * h_2(n) * x_1(n)^2. \end{aligned} \quad (27)$$

3) Third Order: For $m=2$ the constellations are $(k_1k_2)=(12)$ and its permutation (21), and for $m=3$ the only constellation is $(k_1k_2k_3)=(111)$. By Theorem 2 we have

$$\begin{aligned} F_3(z_1,z_2,z_3) = -H_1(z_1z_2z_3)^{-1}[&2H_2'(z_1z_2z_3)F_1(z_1)F_2(z_2,z_3) \\ &+H_3'(z_1z_2z_3)F_1(z_1)F_1(z_2)F_1(z_3)], \end{aligned}$$

and by Theorem 3 we get

$$\begin{aligned} x_3(n) &= -h_1^{-1}(n) * \left(2\sum_i h_2(i)x_1(n-i)x_2(n-i) \right. \\ &\qquad \left. + \sum_i h_3(i)x_1(n-i)^3\right) \\ &= -h_1^{-1}(n) * (2h_2(n) * [x_1(n)x_2(n)] \\ &\qquad + h_3(n) * x_1(n)^3). \end{aligned} \quad (28)$$

4) Fourth Order: Using simplified notation, for $m=2$ the constellations are (13), (31), and (22), for $m=3$ the constellations are (112), (121), and (211), and for $m=4$ the only constellation is (1111). By Theorem 2 we have

$$\begin{aligned} F_4 = -H_1^{-1}[&2H_2'F_1F_3 + H_2'F_2F_2 + 3H_3'F_1F_1F_2 \\ &+H_4'F_1F_1F_1F_1], \end{aligned}$$

and by Theorem 3 we get

$$\begin{aligned} x_4(n) = -h_1^{-1}(n) * \big(&2h_2(n) * [x_1(n)x_3(n)] + h_2(n) * x_2(n)^2 \\ &+3h_3(n) * \left[x_1(n)^2x_2(n)\right] + h_4(n) * x_1(n)^4\big). \end{aligned} \quad (29)$$

5) Fifth Order: For $m=2$ the constellations are (14), (41), (23), and (32), for $m=3$ the constellations are (113), (131), (311), (122), (212), and (221), for $m=4$ the constellations are (1112), (1121), (1211), and (2111), and finally for $m=5$ the only constellation is (11111). By Theorem 2 we have

$$\begin{aligned} F_5 = -H_1^{-1}[&2H_2'F_1F_4 + 2H_2'F_2F_3 + 3H_3'F_1F_1F_3 + \\ &+3H_3'F_1F_2F_2 + 4H_4'F_1F_1F_1F_2 + H_5'F_1F_1F_1F_1F_1], \end{aligned}$$

and by Theorem 3 we get

$$\begin{aligned} x_5(n) = &-h_1^{-1}(n) * \\ &(2h_2(n) * [x_1(n)x_4(n)] + 2h_2(n) * [x_2(n)x_3(n)] \\ &+3h_3(n) * \left[x_1(n)^2x_3(n)\right] + 3h_3(n) * \left[x_1(n)x_2(n)^2\right] \\ &+ 4h_4(n) * \left[x_1(n)^3x_2(n)\right] + h_5(n) * x_1(n)^5). \end{aligned} \quad (30)$$

The expressions in (26), (27), (28), (29), and (30) apply to all types of PWM symmetry (leading-edge, trailing-edge and symmetric) and also for both types of demodulator filter (ideal filter and low-order filter). Two situations where the expressions are simplified should be noted. If the ideal filter is considered, then $H_1(z)=1$ since $h_1(n)=\delta(n)$ as shown in Table I for all type of symmetries. This completely eliminates the filter $h_1^{-1}(n)$ from all the equations. The second situation which simplifies the computation of the prefilter is when considering *symmetric* PWM. For this type of modulation, both $h_m(\cdot)$ and $x_m(\cdot)$ are zero for even values of m, which in turn also simplifies the computation of the odd components $x_3(n), x_5(n), \ldots$.

IV. Experiments: Simulations, Comparisons and Measurements Results

The choice of reconstruction filter will affect the structure of the prefilter as follows: Assuming an ideal low-pass filter, the Volterra kernels have doubly infinite support, and the prefilter must be implemented using (truncated) finite-impulse-response (FIR) filter structures. If a low-order analog low-pass filter is assumed as reconstruction filter, the prefilter can be implemented using infinite-impulse-response (IIR) structures, see [14] for a detailed case study. In the experiments to follow, we choose the FIR structure for implementing the prefilter, corresponding to the ideal filter assumption.

Firstly, the proposed prefilter is simulated and compared to other methods under tough working conditions using very low PWM frequencies: 88.2 kHz and 176.4 kHz: corresponding to $\times 2$ and $\times 4$ up-sampling of the standard 44.1 kHz sampling frequency. For the 88.2 kHz switching frequency the nonlinear behavior of PWM is greatly accentuated and the compensation methods are seriously challenged. Secondly, real measurements from a physical realization of the prefilter using real-time DSP is presented.

A. Comparisons to Other Methods and Simulations

Several methods have been proposed in the literature to reduce PWM distortion. Click modulation is a PWM technique that allows the generation of low switching rate binary signals with separated baseband originally developed using analog signal processing techniques [21]. Implementation of click

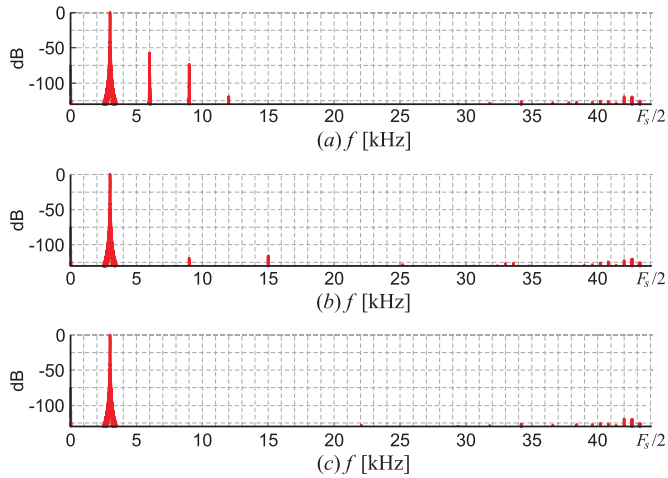

Fig. 6. Simulation with 3 kHz sinusoid. $F_s = 88.2$ kHz. (a) Symmetric PWM. (b) PNPWM. (c) Symmetric PWM with prefilter.

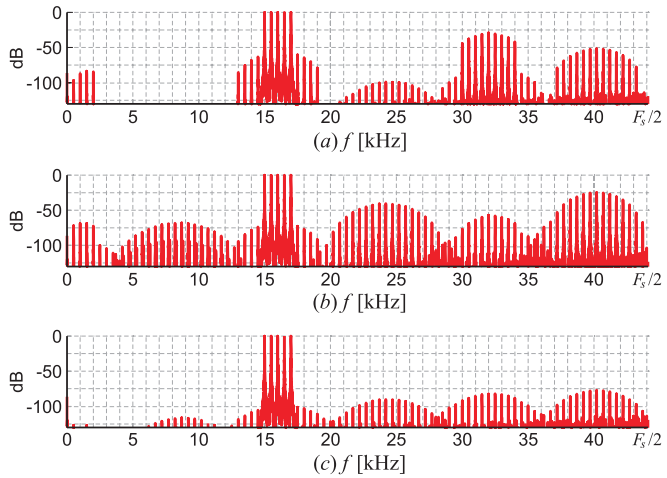

Fig. 7. Simulation with sines of: 15, 15.5, 16, 16.5, and 17 kHz. $F_s = 88.2$ kHz. (a) Symmetric PWM. (b) PNPWM. (c) Symmetric PWM with prefilter.

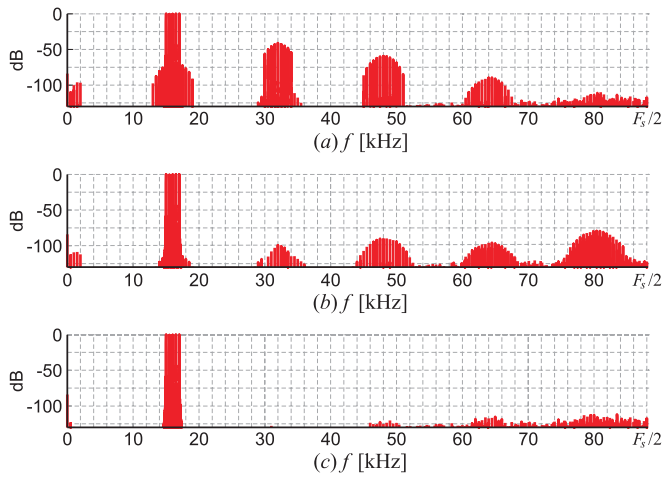

Fig. 8. Simulation with sines of: 15, 15.5, 16, 16.5 and 17 kHz. $F_s = 176.4$ kHz. (a) Symmetric PWM. (b) PNPWM. (c) Symmetric PWM with prefilter.

modulation applied to class-D amplifier are hardware demanding requiring more than one DSP and FPGA and/or they fail to cover the entire audio frequency range [22], [23]. These implementations suffer from the discrete-time approximations applied to implement the digital click modulator [24]. Recently a discrete-time approach of click modulation, which avoids the aliasing problems of the discretization process has been presented [25], but a real-time implementation has not yet been reported.

A widespread method for distortion reduction is pseudo-natural PWM (PNPWM), which is designed to imitate the behavior of analog or natural PWM (NPWM) using digital signal processing. The algorithm is based on a two step procedure: interpolation and crossing-point estimation between the interpolated signal and the carrier signal. If both steps of the algorithm are perfect, then PNPWM will behave as NPWM. It is well known [6], [7] that NPWM avoids harmonics of the modulating signal but does not avoid carrier side-bands to fall into baseband, for this reason the PWM frequency must be high to reduce distortion. This reduces power efficiency because the switching losses of a power stage grows with frequency. Due to its well-known behavior and extensive use, PNPWM was chosen to compare against the proposed prefilter through simulations. Although there are many methods of PNPWM reported in the literature the algorithm proposed by Sarwate *et al.* [5], [6], [17] is one of the better behaved and is chosen for comparisons.

For the simulations Simulink was used. The PWM signal is generated and low-pass filtered with a high-order filter to avoid aliasing when it is sampled to compute the FFT. A 9th order prefilter (for prefilter equations, see [12]) with support $-150 \leq n \leq 150$ was used. Two low PWM frequencies were simulated: 88.2 kHz and 176.4 kHz to show the potential of the proposed algorithm under harsh working conditions. Two input signals, a relatively low-frequency 3 kHz sine and a sum of five high-frequency sines with frequencies between 15 kHz and 17 kHz in steps of 500 Hz were used for the test.

The spectra for $F_s = 88.2$ kHz are depicted in Fig. 6 and Fig. 7. For the low-frequency sinusoidal the behavior of PNPWM and the prefilter is almost the same with only some low distortion components for PNPWM at 9 kHz and 15 kHz. As expected PNPWM can effectively reduce the distortion of this low frequency signal. When a high frequency signal is used, Fig. 7 shows how side-bands of the PWM frequency fall into the baseband [6]. The prefilter reduces all distortion below -100 dB (between 0 Hz and 20 kHz) while PNPWM fails to compensate distortion and is outperformed by standard digital symmetric PWM. Spurious components are also of reduced amplitude between 20 kHz and $F_s/2$ when the proposed method is used, and this eases the process of demodulation. This phenomena is also observed when $F_s = 176.4$ kHz is used, Fig. 8 shows that the proposed method gives a clean spectrum with distortion below -100 dB up to $F_s/2$, again outperforming PNPWM.

Although computationally more demanding, this comparison shows the potential of the proposed approach. The fact that the nonlinear behavior of the digital PWM process is modeled and the prefilter designed to eliminate all distortion (without trying to imitate analog PWM), makes it a more promising technique which allows different operating conditions and not necessary a very high PWM frequency.

B. Measurements Results

A physical implementation of the Volterra prefilter was build and the results are presented in this section. The PWM frequency ("switching frequency") is $F_s = 176$ kHz, a relatively low switching frequency compared to typical class-D amplifier implementations which usually operate at PWM frequencies above 300 kHz, see for example [26, Table I] and citations therein.

1) Symmetric PWM: For this configuration a 5th order prefilter was found to be sufficient to cope with the PWM nonlinearities. The general equations for the model inverses where presented in Section III. For the setup just described, the prefilter equations are reduced to:

$$\begin{aligned} x_1(n) &= u(n), \\ x_3(n) &= -h_3(n) * x_1(n)^3, \\ x_5(n) &= -3h_3(n) * \left[x_1(n)^2 x_3(n)\right] - h_5(n) * x_1(n)^5, \end{aligned} \tag{31}$$

where the expressions for $h_3(n)$ and $h_5(n)$ are given in the "Symmetric" column in Table I. The support of the kernels are truncated to the sample range $-50 \leq n \leq 50$. This length for the FIRs of the prefilter was found to be enough to keep distortion components below -100 dB, for detailed simulations of the prefilter's performance as a function of the FIR support please refer to [12]. Finally, the prefilter output is computed as $\hat{x}(n) = x_1(n) + x_3(n) + x_5(n)$. The computational complexity of the prefilter can be addressed using (31). Three convolutions with symmetric FIR filters with impulse response of length $N = 101$ are computed. This implies $3(N+1)/2$ multiplications, additionally 3 multiplications are required to compute $x_1(n)^3$ and $x_1(n)^2 x_3(n)$ and 1 additional multiplication to compute $x_1(n)^5$ giving a total of 157 multiplications per sample.

The digital input signal is received in S/PDIF format, a standard digital audio protocol used in consumer electronics [27, p. 77]. It is processed with a Sharc ADSP-21469 (450 MHz core clock) digital signal processor (DSP). First, the input signal is up-sampled to the PWM frequency using the sample-rate converter block provided by the DSP and then, the prefiltering takes place. The DSP also provides a PWM block which operates at PWM-clock of 225 MHz and for symmetric PWM the switching frequency should be chosen as an integer divisor of half the PWM-clock. Based on hardware restrictions, a switching frequency of $F_s = 176$ kHz was used for the experiments. Noise feedback coding (NFC) [14] is included to shape the quantization noise out of the audio frequency band (0–20 kHz). All operations are in real-time.

Standard measurements [27] were done to verify the performance of the Volterra prefilter. A high precision signal analyzer (AP-2722 [28]) was used both for generating the input digital test signals $u(n)$ using S/PDIF, and for measuring the results.

Single sinusoid distortion: One of the simplest and most extensively used methods to determine the nonlinear performance of audio amplifiers is to excite the device under test with a pure sinusoidal signal [27]. The input signal used for this test must have energy only at the desired frequency F_1 while the noise and harmonic components should be negligible. The nonlinearities of digital PWM will produce harmonics of different frequencies: $2F_1, 3F_1, \ldots$ and also side-bands of the switching frequency F_s which may fall into the audio frequency band (0–20 kHz) [6]. Frequency measurements with the signal analyzer allows us to record the energy of the harmonics generated by PWM relative to the amplitude of the fundamental component.

Measurements for a $F_1 = 3$ kHz sinusoid are shown in Fig. 9. Fig. 9(a) shows the obtained result when not using any prefilter. The harmonic components at $2F_1$ and $3F_1$ are clearly present, and the amplitudes relative to the fundamental frequency amplitude are -66 dB for $2F_1$ and -77 dB for $3F_1$. These components directly result from the nonlinear behavior of digital PWM.[1] Using the prefilter given in (31), the measured spectrum is depicted in Fig. 9(b). The second harmonic at $2F_1$ and the third harmonic at $3F_1$ are indistinguishable from the noise floor.

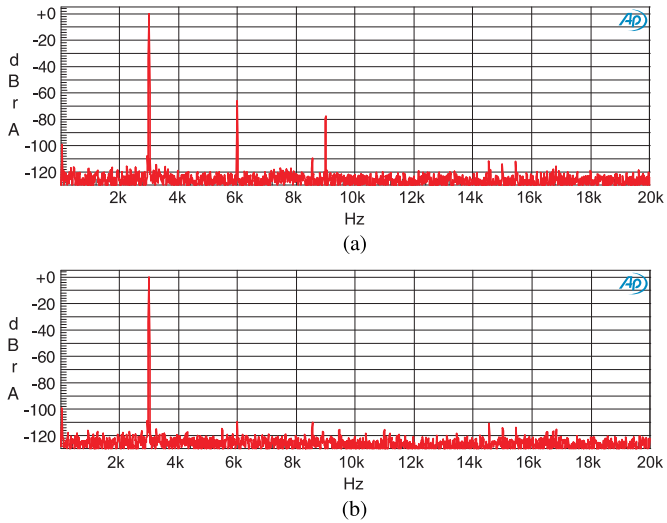

Fig. 9. Symmetric PWM. Single 3 kHz sinusoidal measurement. $F_s = 176$ kHz. (a) Symmetric PWM. (b) Symmetric PWM with prefilter.

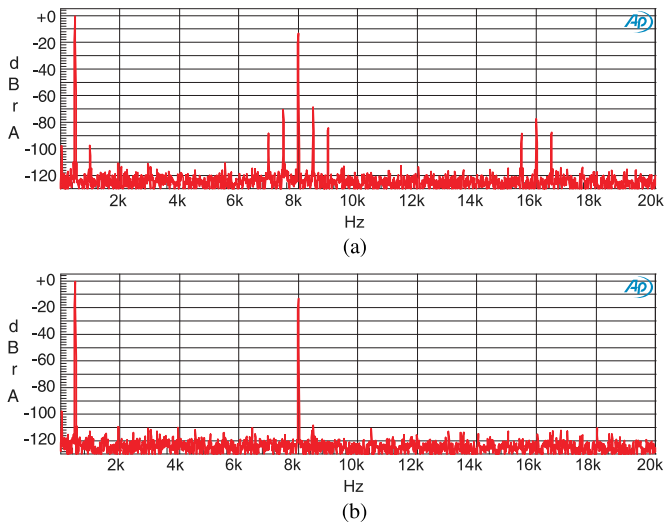

Fig. 10. Symmetric PWM. IMD using 500 Hz and 8 kHz sinusoid. $F_s = 176$ kHz. (a) Symmetric PWM. (b) Symmetric PWM with prefilter.

Intermodulation distortion (IMD): This test signal is the sum of two sinusoidal signals [27, p. 35] of frequencies F_1 and F_2. For the experiment the high frequency component ($F_2 = 8$ kHz) has 1/4 (-12.04 dB) of the amplitude of the lower frequency sinusoid ($F_1 = 500$ Hz).

When the system under test is nonlinear, frequency components at $nF_2 \pm mF_1$ with m and n integers appear. The spectra for the first IMD measurements with and without the prefilter are shown in Fig. 10. When no prefilter is used, intermodulation components appears as shown in the measurement of Fig. 10(a). These unwanted components are at $2F_1$ (≈ -98 dB), $F_2 \pm F_1$ (≈ -70 dB), and $F_2 \pm 2F_1$ (≈ -85 dB). Also at $2F_2$ (≈ -78 dB) and $2F_2 \pm F_1$ (≈ -88 dB). Fig. 10(b) clearly shows how the proposed prefilter eliminates all the intermodulation components.

[1]Even order harmonics like the component at $2F_1$ are generated due to the fact that the input to the Volterra model is of the form $x(n) = 1/2 + A\sin(2\pi(F_1/F_s)n)$ because $0 \leq x(n) \leq 1$ and hence the odd order non-linearities also produce even order harmonics (due to the 1/2 factor).

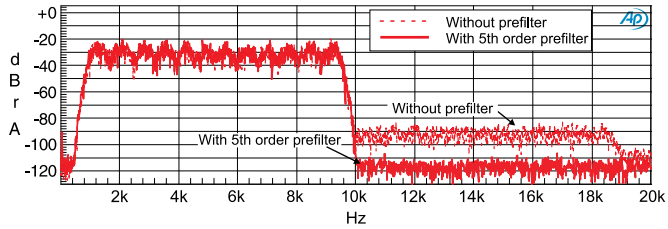

Fig. 11. Symmetric PWM. Band-limited noise. $F_s = 176$ kHz. Superimposed measurements without and with 5th order prefilter.

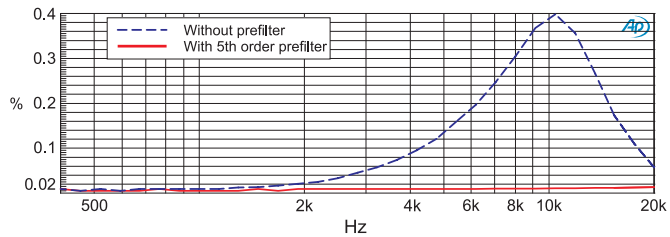

Fig. 12. Symmetric PWM. THD+N Vs. frequency measurement. $F_s = 176$ kHz. Without and with 5th order prefilter.

Band-limited random Gaussian noise: Fig. 11 shows the spectra of the PWM signal for a modulating signal generated by band-limiting white Gaussian noise. The spectra without prefilter and with 5th order prefilter are superimposed to facilitate the comparison. To generate the modulating signal white Gaussian noise is filtered with a band-pass filter with cut-off frequencies of 1 kHz and 9.5 kHz. Without the prefilter residual noise due to the PWM nonlinearity is present in the 10 kHz and 19 kHz frequency band. The proposed prefilter clearly reduces this distortion down to the noise floor. This measurement shows the good behavior of the prefilter under a non-sinusoidal modulating signal.

Total Harmonic Distortion plus Noise (THD+N): Most signal analyzers for audio applications provide a tool to measure the THD+N. This is a quantitative measurement of the complete distortion caused by the device under test and includes not only the harmonic distortion but also any additional noise. It is measured by using a sinusoidal signal (fundamental frequency) as input and then computing the quotient between the energy of the harmonics plus noise (excluding the fundamental) and the energy at the fundamental frequency. Fig. 12 shows the superimposed curves of THD+N versus frequency without the prefilter and with the 5th order prefilter. Without the prefilter the THD+N grows rapidly as frequency increases until reaching its maximum (approximately 0.4%) at around 10 kHz. At this point THD+N decreases because the harmonic at 20 kHz is out of the measurement frequency band. When the prefilter is working the differences are significant, a flat THD+N measurement is obtained in the frequency range of interest.

2) Non-Symmetric PWM: Trailing-Edge: Trailing-edge and leading-edge nonlinearities are equivalent and therefore only trailing-edge is chosen to evaluate the performance of non-symmetric PWM. Due to the slow time decay of the trailing-edge/leading-edge kernels, the support of the prefilter kernels must be much larger than for symmetric PWM. The kernel support is now increased from ± 50 to ± 6000 samples.

Fig. 13 depicts the spectra for the 3 kHz sinusoidal and Fig. 14 for the IMD test. Both experiments were done using $F_s = 176$ kHz and 5th order prefilter, which was also used for the symmetric PWM experiments. The prefilter equations (26), (27), (28), (29) and (30) without the inverse $h_1^{-1}(n)$ are

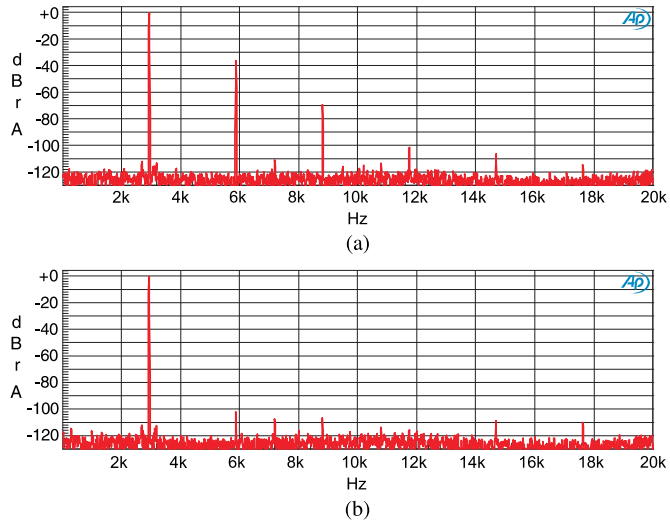

Fig. 13. Trailing-edge PWM. Single 3kHz sinusoidal measurement. $F_s = 176$ kHz. (a) Trailing-Edge PWM. (b) Trailing-edge PWM with prefilter.

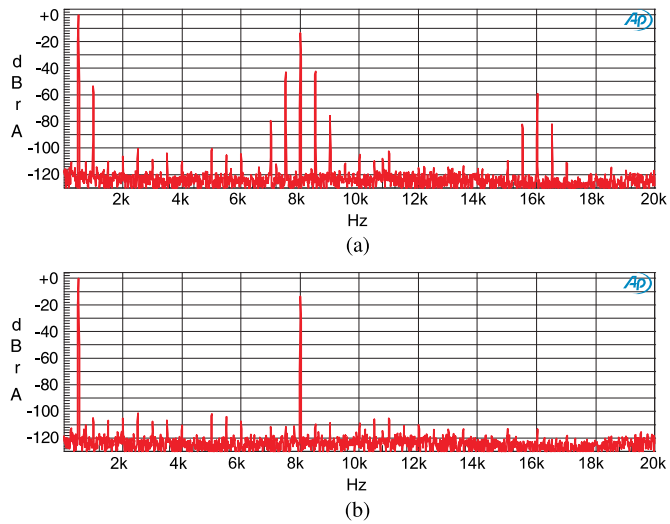

Fig. 14. Trailing-Edge PWM. IMD using 500 Hz and 8 kHz sinusoid. $F_s = 176$ kHz. (a) Trailing-edge PWM. (b) Trailing-edge PWM with prefilter.

implemented. Due to limited DSP processing capabilities the prefiltering for trailing-edge was done offline. This does not affect the experimental results, but is clearly a challenge for implementation of a real-time trailing edge prefilter. Future work should consider IIR solutions as was done in [14].

The spectrum in Fig. 13(a), without prefilter, reveals the accentuated nonlinear behavior of trailing-edge PWM compared to symmetric PWM in Fig. 9(a). The second harmonic at 6 kHz is 30 dB higher than for symmetric PWM while the third harmonic at 9 kHz is 10 dB higher, additionally higher order harmonics can be observed for trailing-edge. Using the 5th order prefilter, Fig. 13(b) reveals that the amplitude of the distortion component at 6 kHz is reduced by more than 65 dB and that all the residual distortion components are below -100 dB. For the IMD test the spectrum without prefilter is shown in Fig. 13(a). Compared with the symmetric case in Fig. 10(a), the amplitude of all distortion components are much higher, this is in agreement with the theoretical analysis presented. When the 5th order prefilter is operating all residual distortion components are below -100 dB as shown in Fig. 13(b).

V. Conclusions

This paper presented for the first time, and in a unified framework, nonlinear Volterra models for three types of PWM symmetries and considering two types of demodulation filters. The mathematical models, which are time-discrete, were expressed using diagonal kernel functions for all cases. It follows from the analysis that trailing and leading edge PWM exhibit more nonlinearities than symmetric PWM. This conclusion is in agreement with previous results which are based on frequency-domain analysis of PWM but in this work the result is presented from a time-domain perspective and verified experimentally. General expressions for the model inverses that enable precompensation of PWM induced distortion were also presented. The performance of the prefilter was evaluated extensively: firstly, simulations and comparisons with pseudo-natural PWM showed the superiority of the proposed approach under tough operating conditions (low PWM frequency); secondly, the prefilter performance was demonstrated experimentally using a digital signal processor. Measurements with a high precision signal analyzer showed that using a 5th order prefilter the distortion is reduced to the noise-floor level even when a relatively low PWM frequency (176 kHz) is used. The prefilter enables a low PWM frequency operation which reduces the power loss of the amplifier. In the experiments it was verified that for symmetric PWM the prefilter can be implemented using short FIR filters, on the other hand, non-symmetric PWM requires a prefilter with long FIRs due to the slow time decay of the Volterra kernels.

References

[1] A. Bindra, "Pulsewidth modulated controller integrated circuit: Four decades of progress [A look back]," *IEEE Power Electron. Mag.*, vol. 1, pp. 10–44, Sep. 2014.

[2] H. S. Black, *Modulation Theory*. New York: van Nostrand, 1953.

[3] M. L. Stephens and J. P. Wittman, "Switched-mode transistor amplifiers," *Trans. Amer. IEE I, Commun. Electron.*, vol. 82, pp. 470–472, Sep. 1963.

[4] M. Rashid, *Microelectronic Circuits: Analysis & Design*. Stamford, CT, USA: Cengage Learn., 2010.

[5] C. Pascual, Z. Song, P. T. Krein, D. V. Sarwate, P. Midya, and W. J. Roeckner, "High-fidelity PWM inverter for digital audio amplification: Spectral analysis, real-time DSP implementation, results," *IEEE Trans. Power Electron.*, vol. 18, no. 1, pp. 473–485, 2003.

[6] Z. Song and D. V. Sarwate, "The frequency spectrum of pulse width modulated signals," *Signal Process.*, vol. 83, no. 10, pp. 2227–2258, 2003.

[7] D. G. Holmes and T. A. Lipo, *Pulse Width Modulation for Power Converters: Principles and Practice*. Hoboken, NJ, USA: Wiley, 2003, vol. 18.

[8] H. Mouton and B. Putzeys, "Understanding the PWM nonlinearity: Single-sided modulation," *IEEE Trans. Power Electron.*, vol. 27, pp. 2116–2128, Apr. 2012.

[9] J. Goldberg and M. Sandler, "Pseudo-natural pulse width modulation for high accuracy digital-to-analogue conversion," *Electron. Lett.*, vol. 27, no. 16, pp. 1491–1492, 1991.

[10] Z. Yu, Y. Fan, L. Shi, and G. Lv, "A pseudo-natural sampling algorithm for low-cost low-distortion asymmetric double-edge PWM modulators," *Circuits, Syst., Signal Process.*, pp. 1–19, 2014.

[11] J. Huang, K. Padmanabhan, and O. Collins, "The sampling theorem with constant amplitude variable width pulses," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, pp. 1178–1190, Jun. 2011.

[12] S. O. Aase, "A prefilter equalizer for pulse width modulation," *Signal Process.*, vol. 92, no. 10, pp. 2444–2453, 2012.

[13] F. Chierchie and E. Paolini, "Real-time digital PWM with zero baseband distortion and low switching frequency," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, pp. 2752–2762, Oct. 2013.

[14] S. O. Aase, "Digital removal of pulse-width-modulation-induced distortion in class-D audio amplifiers," *IET Signal Process.*, vol. 8, pp. 680–692(12), Aug. 2014.

[15] F. Chierchie, E. Paolini, L. Stefanazzi, and A. Oliva, "Simple real-time digital PWM implementation for Class-D amplifiers with distortion-free baseband," *IEEE Trans. Ind. Electron.*, vol. 61, pp. 5472–5479, Oct. 2014.

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

[17] K. Nguyen and D. Sarwate, "Up-sampling and natural sample value computation for digital pulse width modulators," in *Proc. 40th Annu. Conf. Inf. Sci. Syst.*, Mar. 2006, pp. 1096–1101.

[18] G. L. Sicuranza, "Theory and approximation of polynomial filters," *IEEE Circuits Syst. Tutorials, ISCAS*, vol. 94, pp. 50–58, 1996.

[19] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, ser. Signal Processing Series. Upper Saddle River, NJ, USA: Prentice-Hall, 1999.

[20] Y.-W. Fang, L.-C. Jiao, X.-D. Zhang, and J. Pan, "On the convergence of volterra filter equalizers using a pth-order inverse approach," *IEEE Trans. Signal Process.*, vol. 49, no. 8, pp. 1734–1744, 2001.

[21] B. Logan Jr, "Click modulation," *AT&T Bell Lab. Tech. J.*, vol. 63, no. 3, pp. 401–423, 1984.

[22] M. Streitenberger, F. Felgenhauer, H. Bresch, and W. Mathis, "Class-d audio amplifiers with separated baseband for low-power mobile applications," in *Proc. IEEE Int. Conf. Circuits Syst. Commun.*, 2002, pp. 186–189.

[23] K. Sozanski, "Digital realization of a click modulator for an audio power amplifier," *Przeglad Elektrotechniczny*, vol. 86, no. 2, pp. 353–357, 2010.

[24] S. Santi, M. Ballardini, R. Rovatti, and G. Setti, "The effects of digital implementation on zepoc codec," in *Proc. Eur. Conf. Circuit Theory Design*, Aug.–Sep. 2005, vol. 3, pp. 173–176.

[25] L. Stefanazzi, A. Oliva, and E. Paolini, "Alias-free digital click modulator," *IEEE Trans. Ind. Informat.*, vol. 9, no. 2, pp. 1074–1083, 2013.

[26] S.-H. Y. *et al.*, "A dual-level dual-phase pulse-width modulation class-D amplifier with 0.001% THD, 112 dB SNR," in *Proc. IEEE ISCAS*, Jun. 2014, pp. 2676–2679.

[27] B. Metzler, *Audio Measurement Handbook*. Beaverton, OR, USA: Audio Precision, 1993.

[28] A. Precision, *2700 Series User's Manual*. Beaverton, OR, USA: Audio Precision, 2011.

Fernando Chierchie (M'11) received his electronic engineering degree in 2009 and his MS degree in control systems in 2011 both from Universidad Nacional del Sur, Bahía Blanca, Argentina. He is teaching assistant at Digital Signal Processing. His research interests are switching amplifiers, signal processing, and DSP implementations of digital control and signal processing algorithms.

Sven Ole Aase was born in Stavanger, Norway, in 1965. He received the M.Sc. and Ph.D. degrees in 1989 and 1993, respectively, both from the Norwegian Institute of Technology, Trondheim, Norway. Since 2001 he has been a professor at the Department of Electrical engineering and Computer Science, University of Stavanger, Norway. His research interests are signal processing and digital amplifiers.