On the analysis and emulation of nonlinear component characteristics

André Richter, Stefanos Dris, Nuno André
VPIphotonics GmbH, Carnotstr 6, Berlin, Germany, andre.richter@VPIphotonics.com

Abstract: We present theoretical background and implementation considerations of Volterra series based nonlinear component models. We demonstrate the advantage of combining frequency and time domain representations and present an efficient way to find high-order filter taps.
OCIS codes: (060.4510) Optical communications; (250.0250) Optoelectronics; (000.4430) Numerical approximation and analysis

1. Introduction

The performance limits of modern communication systems are often determined by non-ideal characteristics of the transceiver components. With the proliferation of multi-level QAM formats in coherent transmission systems, there is an increasing need for high-performance, linear response electro-optical components. Careful modeling-based emulation of deviations from the non-ideal response characteristics of such components helps to analyze their impact on the targeted system performance, and to determine critical limits that should be accounted for in specifications and possibly addressed by adequate countermeasures. Isolated or cascaded linear impairments, such as bandwidth constraints or driving speed limitations can be readily described by linear filters in the time or frequency domain (TD, FD) at the transmitter or receiver side. The characteristics of nonlinear components (e.g., saturating elements, e-o-e converters, frequency mixers) do not, in the most general case, obey to the superposition principle (multiplicative scaling and additive operation preservation). This makes it more complex to characterize and emulate them. In this work we focus on methods commonly used to describe and analyze weakly nonlinear components and systems. First we classify various models based on the very general Volterra series expansion theory and discuss implementation aspects. Then we demonstrate the advantage of combining TD and FD representations and present an efficient way to determine high-order filter taps.

2. Theoretical background and implementation considerations

In general, the response $y(t)$ of any continuous time-invariant dynamic system to an input $x(t)$ can be approximated according to the Volterra series expansion theory by

$$y(t) = c + \sum_{p=1}^{P} \int_a^b \cdots \int_a^b h_p(\mu_1, \cdots, \mu_p) \prod_{k=1}^{p} x(t-\mu_k)\, d\mu_k \,, \tag{1}$$

with c a constant, and $h_p(\mu_1, \cdots, \mu_p)$ the p-th order Volterra kernel being symmetrical in $\mu_k, 1 \le k \le p$, and P, a, b not necessarily being finite. In the related Wiener representation [1] the individual operators of the sum in Eq. (1) are arranged such that they are independent from each other, and thus can be estimated individually. As this is generally only possible by averaging over all $x(t)$, only convergence in the mean squared sense can be achieved, while the Volterra representation converges uniformly. In practice, depending on the investigated component or system, simplifications of this very general, but quite useless description lead to applicable models, which emulate the specific characteristic of interest such that its impact on system performance can be studied and countermeasures developed. For a list of early papers on systems-analysis applications of functional Volterra series see [2]. Rewriting Eq. (1) for the discrete-time case with maximum system memory M, utilizing the symmetry condition of h_p and setting $c = 0$ (without loss of generality), and an arbitrary delay d_p for non-causal digital filter realizations, yields

$$y_V[n] = \sum_{p=1}^{P} \sum_{\mu_1=0}^{M} \cdots \sum_{\mu_p=\mu_{p-1}}^{M} h_{p,(\mu_1,\cdots,\mu_p)} \prod_{k=1}^{p} x\left[n - \mu_k + d_p\right] . \tag{2}$$

The number of coefficients of the p-th order Volterra kernel $h_{p,(\mu_1,\cdots,\mu_p)}$ with memory M, as introduced in Eq. (2) is given by $(M+p)!/(M!\,p!)$ [3]. Clearly, computational complexity becomes a drawback if kernels of many different or high orders, and/or large memory are employed. Finding the optimum trade-off between complexity reduction (by series truncation and coefficient omission) and the corresponding loss of accuracy represents a demanding design challenge. In what follows we describe a number of important simplifications of the general Volterra series model (for $d_p = 0$ without losing generality). Trivially, Eq. (2) describes a linear filter if $P = 1$, and reduces to a Taylor series for $M = 0$ describing a purely memoryless nonlinear component. A Wiener model is obtained from Eq. (2) by setting $h_{p,(\mu_1,\cdots,\mu_p)} = w_p \prod_{k=1}^{p} h_{\mu_k}$, with w_p polynomial coefficients of the p-th order kernel

$$y_W[n] = \sum_{p=1}^{P} w_p \left(\sum_{\mu=0}^{M} h_\mu x[n-\mu]\right)^p . \tag{3}$$

It can be interpreted as a cascade of a linear filter of memory M and a memoryless nonlinear component of order P, e.g., assuming independence of the frequency and nonlinear characteristics. Finding an adequate Wiener model,

however, is problematic due to the nonlinear dependence on its coefficients h_μ, which complicates accurate adaptive filter realizations. Reversing the order of the memoryless nonlinear component and the linear filter, one obtains the Hammerstein model, represented by

$$y_H[n] = \sum_{\mu=0}^{M} h_\mu \sum_{p=1}^{P} w_p x^p[n-\mu] \,, \tag{4}$$

which is obtained from Eq. (2) by setting $h_{p,(\mu_1,\cdots,\mu_p)} = w_p h_\mu$ for $\mu_1 = \cdots = \mu_p$, and 0 otherwise. Advantageously compared to the Wiener model, the Hammerstein model depends linearly on its coefficients $h_\mu w_p$. Both approaches can be combined by inserting a memoryless nonlinear component between two (possibly different) linear filters to form a more general model structure. Generalizing the Hammerstein model by having $w_p h_\mu = h_{p,\mu}$ in Eq. (4), and considering only mixing terms of type $x[n-\mu]\,|x[n-\mu]|^{p-1}$, e.g., ignoring higher order frequency multiplication terms one obtains a description for the Memory Polynomial model [4]

$$y_{MP}[n] = \sum_{p=1}^{P} \sum_{\mu=0}^{M} h_{p,\mu} x[n-\mu]\,|x[n-\mu]|^{p-1} \,, \tag{5}$$

which can be extended by adding several leading and lagging envelope multiplication terms as desired.

For instance, in [5] the MZMs of an IQ-transmitter are modeled as Wiener system cascading a dynamic linear part (response of electrodes and e-o wave walk-off), and a static nonlinear part (e-o field transfer function of the MZI). The DAC and driver amplifier are modelled by a truncated 3rd-order Volterra series. In [6] the IQ-transmitter components are jointly modeled by a limited memory polynomial considering only the 1st, 3rd and 5th orders. The Indirect-learning architecture [7] is used in both cases to find the coefficients of adequate pre-distortion filters negating the detected non-ideal transfer characteristics. Note that the adaptive sample-based tap optimization of such filters for systems with only weak nonlinearity using stochastic gradient algorithms, such as the least mean-squares (LMS) algorithm, is numerically efficient but shows typically slow convergence [8].

3. Volterra series in time and frequency

Generally, a Volterra filter is simpler to implement in TD compared to its FD realization, especially for time-varying settings that require adaptive structures. It lacks, however, an intuitive understanding of the required number of filter taps and their adaptation. On the other hand, tap measurement for FD Volterra models is relatively straightforward, provided the device is weakly nonlinear: one or several sine waves are fed to the component under test with carefully varied amplitudes to separate the order responses [9]. The implementation, however, is computationally expensive. Combining TD and FD interpretations helps to simplify both, the characterization and implementation of the Volterra filter. For this, the measured n-dim frequency response is converted into the TD via the n-dim Fourier transform (FT) providing information on the n-th order Volterra filter taps.

We investigate the modeling of reference 2nd and 3rd order FD kernels: H_2 and H_3, described by 2- and 3-dim 1st order low-pass transfer functions with 10 GHz corner frequency. Both Volterra kernels are initially realized with a memory of 255, covering a bandwidth of 80 GHz. Investigating the corresponding TD Volterra realization, we find that the energy is concentrated around a few number of taps, suggesting that effective filter tap truncation will not lead to significant performance loss. We explore two simple types of TD tap truncation: applying a square (cube for 3-dim) or a cross of 2 bars (3 bars for 3-dim) inside which all taps are considered, while they are discarded outside of it. We measure truncation with respect to the discarded number of taps and correspondingly reduced energy. The performance is calculated by applying the FT to the truncated TD Volterra taps and comparing the truncated transfer functions $H'_{2,3}$ with the originals $H_{2,3}$ using the metric $\sigma = \sqrt{\langle \Delta H^2 \rangle}$ with $\Delta H^2 = \left| 1 - \left|H'_{2,3}\right|^2 / \left|H_{2,3}\right|^2 \right|$.

Figure 1 shows the FD error σ for selected truncations and depicts its dependence on truncation percentage. For the selected nonlinear component, cross-type truncation allows much larger tap reduction for the same σ compared to the square-type truncation. Further improvements can be obtained by using more sophisticated methods where, for instance, all taps with amplitudes smaller then a certain threshold are discarded, irrespective of their position.

	H_2		H_3	
	90% less taps	76% less taps	88% less taps	62% less taps
Cross-type truncation	**20% lower E** $\sigma = 0.045$	**10% lower E** $\sigma = 0.028$	**20% lower E** $\sigma = 0.028$	**10% lower E** $\sigma = 0.021$
Square-type truncation	33% lower E $\sigma = 0.113$	24% lower E $\sigma = 0.095$	38% lower E $\sigma = 0.116$	22% lower E $\sigma = 0.100$

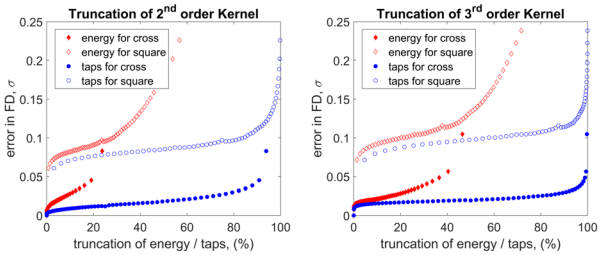
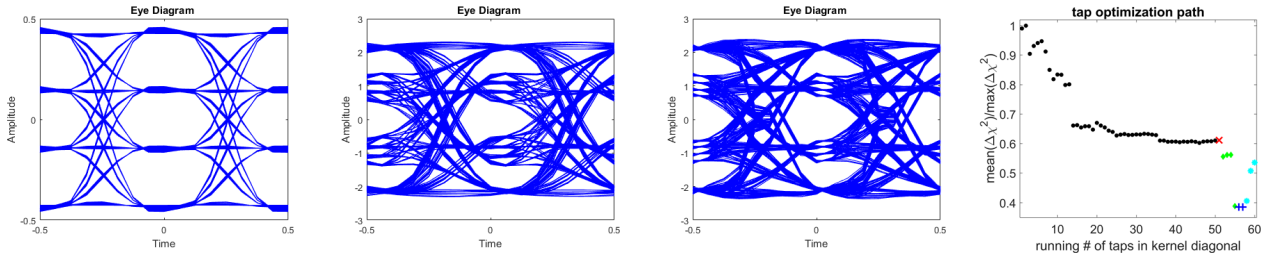


Figure 1: Left: performance (in σ) for selected TD cross- and square-type tap truncations and corresponding energy (E) reductions. Right: σ vs. truncation relative to energy and taps (with a step of four points in each dimension of the truncating mask in TD for H_2 and H_3)

4. Determining the Volterra coefficients

We now discuss the Volterra filter modeling of a modulator driver amplifier, whose early prototype is designed in Keysight ADS, in order to integrate it into system-level simulations of a QAM transmitter (in VPItransmissionMaker Optical Systems) and observe its impact on the output signal quality. A 100 Gbaud PAM-4 training signal is fed to the electronic circuit model of the amplifier to obtain the corresponding output amplified signal. Both signals are then processed using the Least Squares (LS) algorithm to find the Volterra filter kernels that model the nonlinear behavior. For the results shown here, a 640 mV_{pp} swing is used for the input training signal, which ultimately determines the signal swings for which the Volterra model is valid. Ideally, the power spectral density of the training signals should be non-zero up to the Nyquist frequency; here we use a sampling rate of 800 GSa/s (i.e. 8 samples/symbol).

The procedure followed to determine the order, as well as the memories of each kernel order of the Volterra filter is now described. We start with the 1st order, and vary the memory. Each time, the LS algorithm is applied on the input/output training signals, and a set of coefficients is obtained. The LS estimate for each of the above cases yields a specific set of Volterra coefficients, which are then employed to filter a PAM-4 signal having a symbol sequence different to that used for the training. To evaluate the accuracy of each derived Volterra model, we compare the eye diagram of this filtered PAM-4 signal to the eye diagram of the output training signal. A good model should produce an eye diagram which is close to that used in deriving its coefficients, and should do so regardless of the symbol sequence being supplied to it. To quantify this similarity, we compile histograms of the levels at each of the 8 sampling instants of the two eyes, compute the corresponding χ^2 distances [10], and take the mean of these as figure of merit (lower value indicates greater degree of similarity). For the 1st order, increasing the memory beyond a certain value yields negligible improvement. Therefore we fix it, and apply the same procedure to the higher order kernels. We find that an optimal memory value exists, beyond which the accuracy of the model deteriorates. We choose this value each time we proceed with finding the memory of the next higher order. The process is stopped when adding the next higher order will only decrease the accuracy of the model. In our example, we obtain the best result with a 4th order Volterra filter with 50, 1, 4 and 2 taps in the respective kernel diagonals (Figure 2).



(a) (b) (c) (d)

Figure 2: Input (a) and output (b) training signals, and output signal of Volterra filter (c). Volterra tap optimization path (d), where different symbols indicate different kernel orders, and the last ones (cyan *) belong to the discarded 5th order kernel.

5. Conclusion

We discussed various flavors of Volterra series based nonlinear component models, put them into perspective with each other and elaborated on implementation and accuracy constraints. We showed that combining the information of frequency and time domain nonlinear modeling simplifies the characterization and implementation of a 2nd and 3rd order kernel. Finally, we presented an efficient procedure for determining the filter orders and their corresponding memories required for accurate Volterra series fitting of the electrical circuit model of a modulator driver amplifier.

6. References

[1] N. Wiener, "Nonlinear Problems in Random Theory," Wiley, New York, 1958.
[2] J.F. Barrett, "Bibliography of Volterra series, Hermite …" EE Department, Eindhoven Univ. of Technology, NL 1977, T-H report 77-E-71.
[3] E.L.O. Batista, R. Seara, "A reduced-rank approach …," EURASIP Journal on Advances in Signal Processing, vol. 18, 2016: 118, 2016.
[4] D.R. Morgan, et al., "A Generalized Memory …," IEEE Transactions on Signal Processing, vol. 54, no. 10, pp. 3852-3860, 2006.
[5] P.W. Berenguer, et al., "Nonlinear digital pre-distortion …," Journal of Lightwave Technology, vol. 34, no. 8, pp. 1739-1745, 2016.
[6] G. Khanna, et al., "A Robust Adaptive Pre-Distortion Method …," IEEE Photonics Technology Letters. vol. 28, pp. 752–755, 2016.
[7] C. Eun, E.J. Powers, "A new Volterra predistorter based on …," IEEE Transactions on Signal Processing, vol. 45, no. 1, pp. 223-227, 1997.
[8] P.S. Diniz, "Nonlinear Adaptive Filtering," in: Adaptive Filtering. Springer, Boston, MA, 2008.
[9] S. Boyd, Y. Tang, L. Chua, "Measuring Volterra kernels," IEEE Transactions on Circuits and Systems, vol. 30, no. 8, pp. 571-577, 1983.
[10] *NIST/SEMATECH e-Handbook of Statistical Methods*, Chapter 1.3.5.15., https://www.itl.nist.gov/div898/handbook/, Oct 2018.

Acknowledgement: This work was supported through the EU H2020 project QAMeleon (780354) and the German Ministry of Education and Research in frame of the Celtic project SENDATE-FICUS (16KIS0489). The authors thank Romain Hersent and Jean-Yves Dupuy of III-V Lab for providing the modulator driver data used to obtain the Volterra coefficients.