

Improving the approximation ability of Volterra series identified with a cross-correlation method

Simone Orcioni

Received: 10 December 2013 / Accepted: 31 July 2014 / Published online: 16 September 2014
© The Author(s) 2014. This article is published with open access at Springerlink.com

Abstract This paper proposes an improvement in cross-correlation methods derived from the Lee–Schetzen method, in order to obtain a lower mean square error in the output for a wider range of the input variances. In particular, each Wiener kernel is identified with a different input variance and new formulas for conversion from Wiener to Volterra representation are presented.

Keywords Lee–Schetzen method · Volterra series · Nonlinear systems · Cross-correlation methods

1 Introduction

Volterra series is a polynomial functional series for nonlinear system representation and identification. In [4], Fréchet showed that a sum of a finite number of terms of Volterra series can approximate continuous real-valued mappings, defined on compact aggregates, e.g., on functions defined over intervals. This is an important limitation when forcing signals are defined over an infinite (or semi-infinite) time interval (i.e., periodic

stimuli). A useful property of real systems, which overcomes this limit, is fading-memory. Introduced in the 1930s by Volterra himself [18] can be defined by his own words: “A first extremely natural postulate is to suppose that the influence of the heredity corresponding to states a long time before the given moment gradually *fades out*; ... This postulate represents the principle of the dissipation of hereditary action.” After half a century, it was formalized by Boyd and Chua [1] and Sandberg [15], with slightly different names and definitions. Using this property, the approximation can be extended to continuous operators with input functions defined in an unlimited time interval. Fading-memory or approximately finite memory allows us to approximate arbitrary well a system with truncated Volterra series of finite memory, particularly suitable for discrete time, digital implementations, such as those achievable in practice.

The identification of Volterra series can be carried out by searching the minimum of the mean square error (MMSE) between the outputs of the series and the target system. If the input is taken from an independent identically distributed (i.i.d.) sequence, it is well known that the cross-correlation method due to Lee–Schetzen [16] gives the optimal solution in the MMSE sense. This method needs the output to be expressed as a sum of orthogonal functional as those proposed by Wiener [19], since Volterra functional are not orthogonal to each other.

To overcome the difficulties with gaussian inputs and the high number of free parameters, a common

The supplementary material regards previous work by the author on Volterra series identification.

Electronic supplementary material The online version of this article (doi:10.1007/s11071-014-1631-7) contains supplementary material, which is available to authorized users.

S. Orcioni (✉)
Department of Information Engineering, Università
Politecnica delle Marche, Ancona, Italy
e-mail: s.orcioni@univpm.it

approach to kernel identification is to expand the kernel with a set of basis functions. In [2], a wavelet-basis expansion approach is proposed to identify Volterra kernels. The kernels are estimated by expanding them with four-order B-spline wavelets. In [2], a good review on expansion methods can also be found.

Compared to expansion methods, the Lee–Schetzen method has the possibility of identifying a kernel value independently of the other values. This allows for possibility of testing the memory extent of a kernel without having the need to identify the whole kernel.

The main drawback of this method is the great inaccuracy in the identification of the elements of kernels with at least two equal indices, called diagonal elements. Effective solutions that overcome this problem have been proposed in literature, for series up to the third order in [5], and for a generic order in [14], where a comparison between the two methods can also be found. Using cumulants instead of input moments in [7], Koukoulas at alii proposed a proof of the n th-order case valid also for non-i.i.d. inputs. In the white Gaussian input case, the general formulas in [7] can be shown to reduce to the method proposed in [5].

Another orthogonal method, which does not require the input be an i.i.d. sequence, has been a result of M. Korenberg’s: the Fast Orthogonal Algorithm (FOA) [6]. A comparison between the FOA and Lee–Schetzen methods can be found in [13]. Other identification methods and a survey of possible applications of Volterra series ranging from signal processing to identification of mechanical and physiological systems can be found in [9, 10, 17].

A potential disadvantages of the Volterra theory, as explicitly stated in [17], includes input amplitude limitations related to convergence issues and the need for higher-order kernels. The effect of input amplitude on convergence was investigated in [12], where it has been shown that identification input non-idealities makes the output MSE a function of the input variance.

This work proposes a change in Lee–Schetzen’s method, taking the formulas presented in [14] as the reference for the method, with the aim to improve the approximation capabilities of Volterra series versus errors due to input non-idealities and model truncation. In particular, the proposed method will ensure a lower error for a wider range of input variances and will make it easier to identify high-order kernels.

Section 2 introduces orthogonal methods based on cross-correlation, and in Sect. 3, the new algorithm is

proposed, while in Sect. 4, a comparison between the new algorithm and the reference is made.

2 Orthogonal methods for discrete time Volterra series identification

It is common practice to use the discrete time version of the Volterra series. The output of the discrete time Volterra series can be defined by means of the Volterra series operator

$$y(n) = (Vx)(n) = h_0 + \sum_{i=1}^P (H_i x)(n) \tag{1}$$

$$(H_i x)(n) = \sum_{\tau_1=0}^M \cdots \sum_{\tau_i=0}^M h_i(\tau_1, \dots, \tau_i) \prod_{j=1}^i x(n - \tau_j) \tag{2}$$

where $M \in \mathbb{Z} \cup \{+\infty\}$ e $P \in \mathbb{N} \cup \{+\infty\}$. In (2), the summations start from zero because we are considering causal systems. $h_i(\tau_1, \dots, \tau_i)$, h_0 are called Volterra kernels.

In practical implementation, we always consider M and P as finite and, without loss of generality, the kernels $h_i(\tau_1, \dots, \tau_i)$ as symmetrical. In fact, for the commutativity of the multiplication, it is always possible to symmetrize $h_i(\tau_1, \dots, \tau_i)$ without changing $(H_i x)(n)$. As such, we can indicate a finite P th order Volterra series as $(V^P x)(n)$ and the corresponding i th order operators, with $i = 1, \dots, P$ as

$$(H_i x)(n) = \sum_{\tau_1=0}^M \sum_{\tau_2=\tau_1}^M \cdots \sum_{\tau_i=\tau_{i-1}}^M h_i(\tau_1, \dots, \tau_i) \times \prod_{j=1}^i x(n - \tau_j)$$

Once established that a particular class of systems can be approximated arbitrarily well with truncated Volterra series, the problem of estimating it comes up. To this aim, we will use Wiener’s orthogonal representation [19] and its estimation by means of the cross-correlation method by Lee and Schetzen [8].

2.1 Orthogonalization

In order to allow identification by a cross-correlation method, the Volterra series must be rearranged in terms of orthogonal operators:

$$y(n) = \sum_i (H_i x)(n) \equiv \sum_i (G_i x)(n) \tag{3}$$

These operators, commonly known as Wiener G–functionals, are orthogonal to each other and also to Volterra operators of lower orders if the input $x(n)$ is a stationary white noise with zero mean and variance σ_x^2 . By means of the Gram-Schmidt orthogonalization method, the following expressions for the Wiener G–functional can be obtained:

$$\begin{aligned} (G_0x)(n) &= k_0 \\ (G_1x)(n) &= \sum_{\tau_1} k_1(\tau_1)x(n - \tau_1) \\ (G_2x)(n) &= \sum_{\tau_1, \tau_2} k_2(\tau_1, \tau_2)x(n - \tau_1)x(n - \tau_2) \\ &\quad - A \sum_{\tau_1} k_2(\tau_1, \tau_1) \\ (G_3x)(n) &= \sum_{\tau_1, \tau_2, \tau_3} k_3(\tau_1, \tau_2, \tau_3)x(n - \tau_1)x(n - \tau_2)x(n - \tau_3) + \\ &\quad - 3A \sum_{\tau_1} \sum_{\tau_2} k_3(\tau_1, \tau_2, \tau_2)x(n - \tau_1) . \end{aligned}$$

As can be seen, the Wiener G–functional are non-homogeneous operators.

The Volterra kernels of a truncated series can be derived from the Wiener kernel by means of simple equivalence formulas. In the following, we have reported the Wiener–Volterra formulas valid in the case of a fifth-order Volterra system:

$$h_5 = k_5 \tag{4}$$

$$h_4 = k_4 \tag{5}$$

$$h_3 = k_3 - 10A \sum_{\tau_4} k_5(\tau_1, \tau_2, \tau_3, \tau_4, \tau_4) \tag{6}$$

$$h_2 = k_2 - 6A \sum_{\tau_3} k_4(\tau_1, \tau_2, \tau_3, \tau_3) \tag{7}$$

$$\begin{aligned} h_1 &= k_1 - 3A \sum_{\tau_2} k_3(\tau_1, \tau_2, \tau_2) \\ &\quad + 15A^2 \sum_{\tau_2, \tau_3} k_5(\tau_1, \tau_2, \tau_2, \tau_3, \tau_3) \end{aligned} \tag{8}$$

$$\begin{aligned} h_0 &= k_0 - A \sum_{\tau_1} k_2(\tau_1, \tau_1) \\ &\quad + 3A^2 \sum_{\tau_1, \tau_2} k_4(\tau_1, \tau_1, \tau_2, \tau_2) \end{aligned} \tag{9}$$

These formulas, although well known in literature, are presented for comparison with those proposed later in the paper.

2.2 Cross-correlation

In [14], authors proposed a formula for an efficient implementation of the Lee–Schetzen method for the estimation of Wiener series of any order. The formula simplifies to the classic Lee–Schetzen one if non-diagonal terms of the Wiener kernel are identified.

In the case of diagonal kernel terms identification, the sum-product operator adds, to classic Lee–Schetzen terms, proper values taken from lower-order kernels:

$$\begin{aligned} k_p(\tau_1, \dots, \tau_p) &= \frac{1}{p!A^p} \left\{ E \left\{ y(n) \prod_{i=1}^p x(n - \tau_i) \right\} + \right. \\ &\quad \left. - \sum_{h=1}^{\lfloor p/2 \rfloor} (p - 2h)!A^{p-h} \sum_{\Pi} k_{p-2h}(\tau_{i_1}, \dots, \tau_{i_{p-2h}}) \right\} \tag{10} \end{aligned}$$

where $\lfloor \cdot \rfloor$ denotes the integer part and \mathbf{P} is a set of p integers. The new operator \sum_{Π} is defined as

$$\sum_{\Pi} k_q(\tau_{Q_i}) \triangleq \sum_{i=1}^{\binom{p}{q}} k_q(\tau_{Q_i}) A^{-\frac{q}{2}} E \left\{ \prod_{i=1}^r x(n - \tau_{j_i}) \right\}$$

where $Q_i \subset P$ are all the subsets generated by the combinations of q elements chosen from \mathbf{P} , the τ_j belongs to the complementary set of Q_i with respect to P , so $q + r = p$, and $\tau_{Q_i} = (\tau_{i_1}, \dots, \tau_{i_q})$.

In [12], the same authors show how input non-idealities can affect kernel estimation. In particular, explicit expression of residuals up to the sixth order are given. These residuals occur when there is a difference between the moments of an ideal input and those produced by a pseudo-random finite length input. The residual causes an error in system identification even if the target system itself is a Volterra series of the same order of the model. Tables 1 and 2 show, respectively, the mean values and standard deviation estimate of moments of different orders. The mean value of odd order moments should be zero while that of even order rises exponentially. Table 2 shows how for higher-order moments the standard deviation has acceptable values only for very long inputs. However, also for maximum input length presented in the Tables, an error of 1% in input moments estimation, as present in the highest order, can cause relevant error in kernel estimation.

Another important source of error is the truncation errors that arise if the order of the Volterra series is lower than that of the system to be identified. Computational errors need to be added also.

Table 1 Mean of 10^4 estimate of Moments of different orders, defined as $m_P = \frac{1}{N} \sum_i^N x(i)^P$

P\N	10^3	10^4	10^5	10^6
2	1.000e+00	1.000e+00	1.000e+00	1.000e+0
3	-1.441e-03	3.554e-04	-5.871e-05	6.6354e-05
4	3.003e+00	3.000e+00	3.000e+00	2.9999e+00
5	-4.672e-03	1.727e-03	-1.161e-03	4.6270e-04
6	1.504e+01	1.500e+01	1.500e+01	1.4999e+01
7	2.261e-02	1.642e-02	-1.903e-02	0.00259
8	1.056e+02	1.049e+02	1.050e+02	105.02624

Table 2 Standard deviation of 10^4 estimate of moments of different orders, defined as $m_P = \frac{1}{N} \sum_i^N x(i)^P$

P\N	10^3	10^4	10^5	10^6
2	4.492e-02	1.418e-02	4.456e-03	0.0014216
3	1.227e-01	3.864e-02	1.220e-02	0.0039071
4	3.103e-01	9.708e-02	3.115e-02	0.0097507
5	9.818e-01	3.063e-01	9.688e-02	0.0305327
6	3.188e+00	9.955e-01	3.227e-01	0.1002482
7	1.168e+01	3.661e+00	1.164e+00	0.36568
8	4.418e+01	1.396e+01	4.544e+00	1.44116

For all these reasons, we expect that the MSE of a Volterra series, expressed as a function of the input variance, will have a minimum at the variance used in the identification process and will rise elsewhere, giving rise to a phenomenon that we can call *model locality*.

2.3 Locality of truncated Volterra series

In order to test the previous statement, we identified a nonlinear system of infinite order with a second-order Volterra series, using (10) and a LMS identification algorithm, in the implementation reported in [3]. Due to the difficulties in implementing high-order kernel identification with LMS, in this section, we will show locality of Volterra series only with second-order series, while in the next section, we will show implementation of our algorithm until the fifth order.

The nonlinear system chosen as test is a Wiener model and its diagram is shown in Fig. 1. It is formed of a cascade of a linear filter with a static nonlinearity. The nonlinearity is given by the following function

$$g(x) = \frac{4.5}{1 + 2e^{-x}} - \frac{4.5}{3} \tag{11}$$

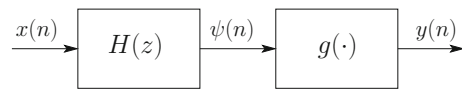


Fig. 1 Schematic of the nonlinear system used in the experiments

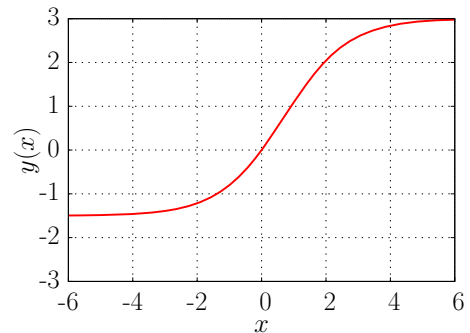
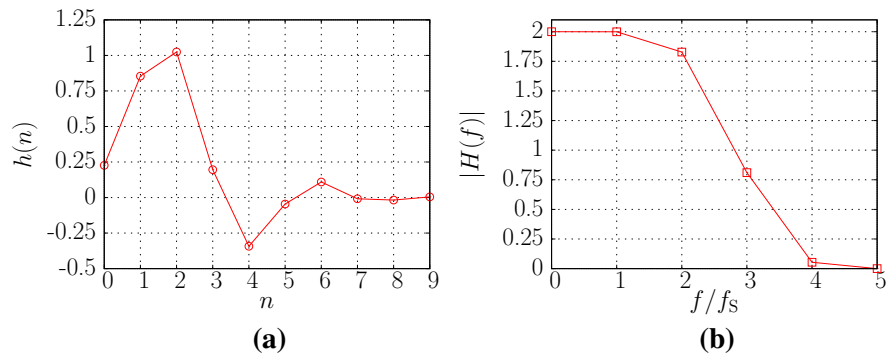


Fig. 2 Characteristic of the static nonlinearity $g(\cdot)$

represented in Fig. 2. This function has a Taylor expansion with rational coefficient, shown in the following formula up to the fifth order:

$$\tilde{g}(x) = x + \frac{9}{54}x^2 - \frac{27}{486}x^3 - \frac{135}{5832}x^4 + \frac{189}{87480}x^5 \tag{12}$$

Fig. 3 **a** Impulsive response of linear filter used in Fig. 1, **b** magnitude of spectrum of the linear filter ($|H(j\omega)|$)



With this expansion, it is easy to generate Volterra systems of finite order with known kernels.

The linear part of the test system is a low-pass filter given by the scaling function of the Daubechies Wavelet of order ten (D10). Figure 3 shows the impulse response and the magnitude of the spectrum of the filter.

Once the test system was established, we used (10) to estimate four second-order Volterra systems that approximates it, each estimated with an input of different variance. These Gaussian inputs had standard deviations $\hat{\sigma}_x$ from the set $\{0.3, 0.6, 1.2, 2.4\}$ and a length of 10^6 samples.

In the LMS algorithm implementation, 10^4 iterations are repeated for one hundred different input signals, each 10^4 samples long for a total of 10^6 weight updating. For stability problems, with higher input variances, the updating coefficient needed to be reduced.

In order to test the identified Volterra series, inputs with standard deviation σ_x in the interval $[0.3/4, \dots, 2.4 \cdot 4]$ and length of $5 \cdot 10^5$ samples were used. The relative root means square errors (RRMSE) between the outputs of the test system and of the Volterra series were calculated, according to the following formula

$$RRMSE = \sqrt{\frac{\sum_{i=1}^N (y(n) - (V^P)(n))^2}{\sum_{i=1}^N y(n)^2}} \quad (13)$$

where $y(n)$ is the output of the test system, $(V^P)(n)$ the output of a P th Volterra series and $N = 5 \cdot 10^5$. The results are shown in Fig. 4.

As expected, each Volterra system has a minimum of the error correspondingly to the standard deviation used in identification. Elsewhere, the error grows rapidly. This makes the Volterra model optimum only locally, independently from the algorithm chosen for the iden-

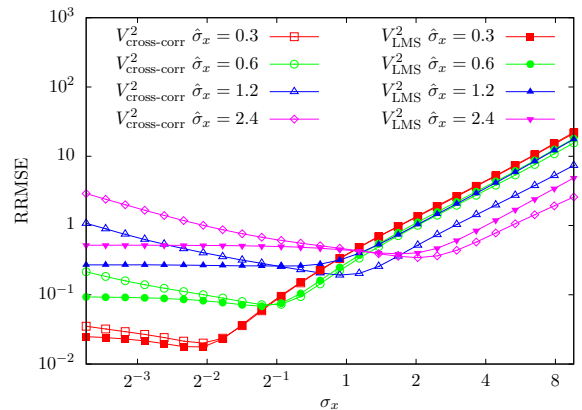


Fig. 4 Relative root mean square error of the output of second-order Volterra series versus the desired output of an ∞ order test nonlinear system

tification. Furthermore, the value of the error corresponding to the minimum increases as the variance of the input used in identification increases. This behavior is due to the cause of errors that we have previously summarized: input non-ideality and order truncation.

3 Proposed algorithm

The aim of the proposed algorithm is to lower the MSE of Volterra series when signals with variances different than that used in the identification are in input to the system. To do this, we use an input with a different variance for each Wiener kernel to be identified.

In the traditional orthogonal algorithm, using high σ_x inputs has the advantage of stimulating high-order nonlinearity, so as to achieve more accurate high-order kernel identification. As a drawback, the use of high σ_x values causes high identification error in lower-order kernels, as can be seen in (4–9) where high-order

Wiener kernels are summed in lower-order Volterra kernel multiplied by variances of lower-order kernels.

The rationale should be to use a low σ_x value for the lower-order kernel and to gradually increase it for those of higher order. This is not a theoretical problem in Wiener kernel identification, since the Wiener functional are orthogonal to each other, but an appropriate normalization is needed in Wiener to Volterra conversion formulas for taking into account the use of different variances. New Wiener to Volterra conversion formulas are needed and will be proposed in the following.

The definition of the new algorithm begins with the rewriting of (10). In particular, the input variances, used for each Wiener kernel, will be indicated as $A_j = \hat{\sigma}_{x_j}^2$

$$k_p^{(j)}(\tau_1, \dots, \tau_p) = \frac{1}{p!A_j^p} \left\{ E \left[y^{(j)}(n) \prod_{i=1}^p x^{(j)}(n - \tau_i) \right] + \sum_{h=1}^{\lfloor p/2 \rfloor} (p-2h)!A_j^{p-h} \sum_{\Pi} k_{p-2h}^{(j)}(\tau_{i_1}, \dots, \tau_{i_{p-2h}}) \right\} \quad (14)$$

where $k_p^{(j)}$ and $y^{(j)}(n)$ are obtained with the input $x^{(j)}(t)$ of variance A_j . In the following, (14) is made explicit for the first three orders:

$$\begin{aligned} k_0^{(0)} &= E\{y^{(0)}(n)\} \\ k_1^{(1)}(\tau_1) &= \frac{1}{A_1} E \left\{ y^{(1)}(n) x^{(1)}(n - \tau_1) \right\} \\ k_2^{(2)}(\tau_1, \tau_2) &= \frac{1}{2!A_2^2} \left\{ E \left[y^{(2)}(n) \prod_{i=1}^2 x^{(2)}(n - \tau_i) \right] - A_2 k_0^{(2)} \delta_{\tau_1 \tau_2} \right\} \\ k_3^{(3)}(\tau_1, \tau_2, \tau_3) &= \frac{1}{3!A_3^3} \left\{ E \left[y^{(3)}(n) \prod_{i=1}^3 x^{(3)}(n - \tau_i) \right] - A_3^2 \left[k_1^{(3)}(\tau_1) \delta_{\tau_2 \tau_3} + k_1^{(3)}(\tau_2) \delta_{\tau_1 \tau_3} + k_1^{(3)}(\tau_3) \delta_{\tau_1 \tau_2} \right] \right\}. \end{aligned}$$

As can be clearly seen in the explicit formulas, the drawback with respect to the classic formula is that for the identification of a p -order kernel, all lower kernels must be identified again with the higher variance. However, an outstanding improvement in the output MSE will be obtained if the Volterra kernels are obtained

with the new Wiener to Volterra formulas:

$$h_5 = k_5^{(5)} \quad (15)$$

$$h_4 = k_4^{(4)} \quad (16)$$

$$h_3 = k_3^{(3)} - 10A_3 \sum_{\tau_4} k_5^{(5)}(\tau_1, \tau_2, \tau_3, \tau_4, \tau_4) \quad (17)$$

$$h_2 = k_2^{(2)} - 6A_2 \sum_{\tau_3} k_4^{(4)}(\tau_1, \tau_2, \tau_3, \tau_3) \quad (18)$$

$$\begin{aligned} h_1 &= k_1^{(1)} - 3A_1 \sum_{\tau_2} k_3^{(3)}(\tau_1, \tau_2, \tau_2) \\ &\quad + 15A_1^2 \sum_{\tau_2} \sum_{\tau_3} k_5^{(5)}(\tau_1, \tau_2, \tau_2, \tau_3, \tau_3) \end{aligned} \quad (19)$$

$$\begin{aligned} h_0 &= k_0^{(0)} - A_0 \sum_{\tau_1} k_2^{(2)}(\tau_1, \tau_1) \\ &\quad + 3A_0^2 \sum_{\tau_1} \sum_{\tau_2} k_4^{(4)}(\tau_1, \tau_1, \tau_2, \tau_2) \end{aligned} \quad (20)$$

These formulas are valid for a fifth-order Volterra series. By comparing them with (4–9), we note that in each Volterra kernel, the diagonal Wiener kernel of higher order are multiplied with the variance corresponding to the order of the Volterra kernel instead of the Wiener kernel.

4 Experimental results

As previously discussed, the main causes of error in the identification of Volterra series are order truncation and input non-ideality. The following experiments were arranged in order to highlight the effect of the new algorithm on each cause separately.

In the first experiments, the truncation error was excluded by choosing, as test systems, Volterra systems of finite order that can ideally approximated with no error with Volterra series of the same order. These Volterra test systems were obtained by truncating the Taylor expansion of $g(\cdot)$, presented in (12), at the desired order. In this way, we obtained Volterra systems with known kernels. Clearly, $h_1(\tau_1)$ will be the impulse response of the filter $h(n)$, represented in Fig. 3a, while the higher-order kernels can be easily obtained by using $h(n)$ and the Taylor expansion coefficient reported in (12).

Figure 5 shows a section of the third-order Volterra kernel, $h_3(8, \tau_2, \tau_3)$. Taylor expansion of $g(\cdot)$, trun-

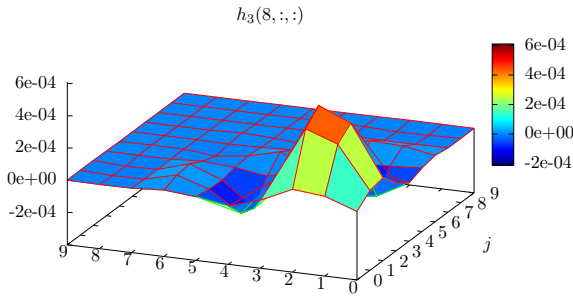


Fig. 5 A section of the Volterra kernel of third order

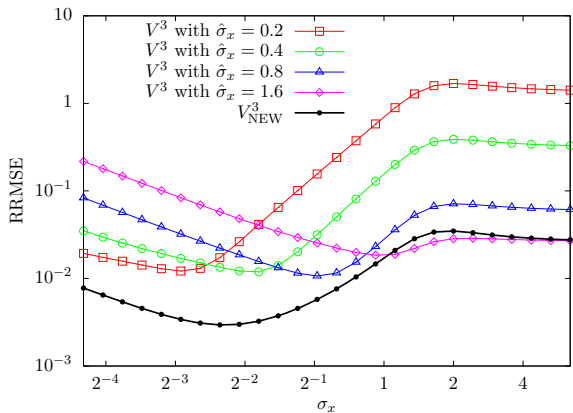


Fig. 6 Relative root mean square error of the output of third-order Volterra series versus the desired output of a third-order test Volterra system

cated at the p -order, will be referred to in the following as $\tilde{g}_p(s)$.

To obtain Fig. 6, four third-order Volterra series (V^3) were used to identify the test system obtained cascading $h(n)$ with $\tilde{g}_3(x)$. Four Volterra series were identified with the classic algorithm (10), using the standard deviations $\hat{\sigma}_x$ in the set $\{0.2, 0.4, 0.8, 1.6\}$. One series (V^3_{NEW}) was estimated with the proposed algorithm (14) using each $\hat{\sigma}_x$ in the same set for each different order kernel. Figure 6 shows the RRMSE between the Volterra series and the test system, calculated accordingly to (13) and obtained with a different input sequence with respect to that used for the identification. Figure 6 shows that the proposed algorithm extends the input region in which the error has acceptable values, dramatically.

The same experiment was carried out with Volterra series of fourth order and a test system obtained cascading $h(n)$ and $\tilde{g}_4(x)$. Figure 7 shows the obtained results. Also with the fourth order, the proposed algorithm

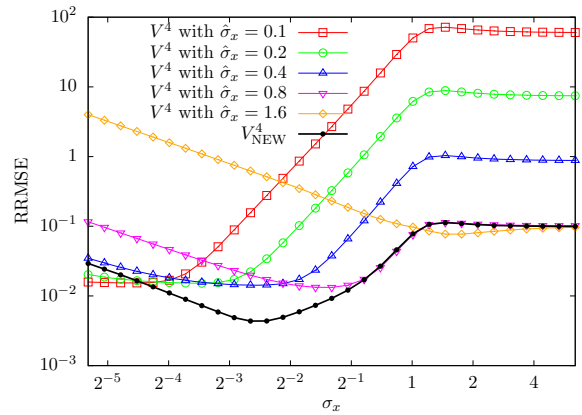


Fig. 7 Relative root mean square error of the output of fourth-order Volterra series versus the desired output of a fourth-order test Volterra system

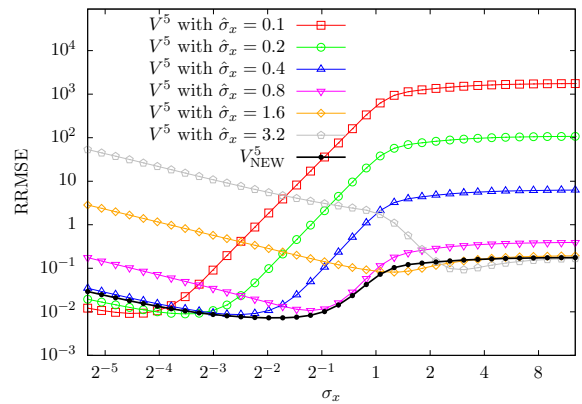


Fig. 8 Relative root mean square error of the output of fifth-order Volterra series versus the desired output of a fifth-order test Volterra system

obtains a considerable extension of the input region in which the error is acceptable. In the same input region, the other Volterra series present errors three order of magnitude greater. It can be noticed that increasing the order of the system the errors obtained with the classic algorithm increase further. This is better shown by Fig. 8, where the results obtained with fifth-order systems have been reported. This Figure shows that, while the relative error obtained with the new algorithm goes slightly above 0.1, the RRMSE obtained with the classic algorithm goes above 10^3 .

So far, the Volterra series was used to identify a nonlinear system of the same polynomial order. Now, the Volterra series will be used to identify the nonlinear system of infinite order, represented in Fig. 1 with the

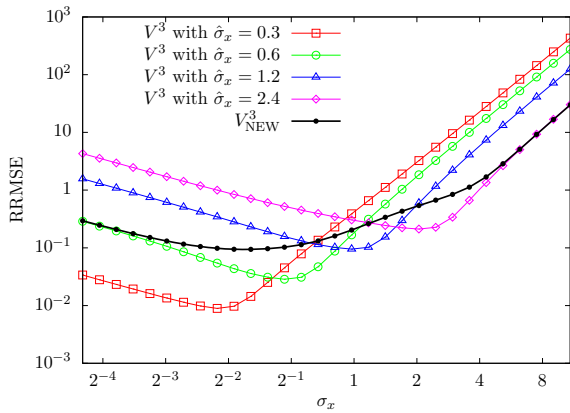


Fig. 9 Relative root mean square error of the output of third-order Volterra series versus the desired output of an ∞ order test Volterra system

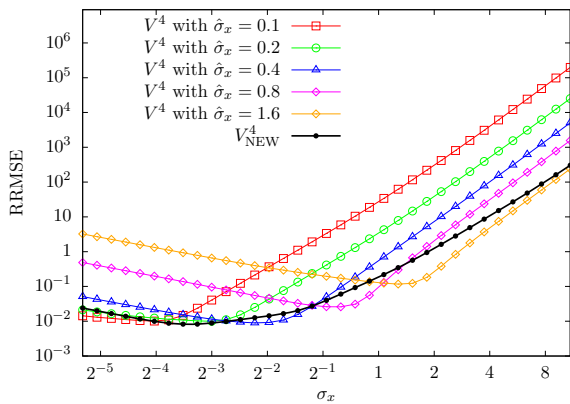


Fig. 10 Relative root mean square error of the output of fourth-order Volterra series versus the desired output of an ∞ order test Volterra system

$g(x)$ function expressed by (11). In this way, the truncation error was added to the other error sources. For all the experiments with an infinite order test system, the input length was increased to $4 \cdot 10^6$ samples.

Figure 9 shows the results obtained with the third-order Volterra series. The truncation error behaves differently from the previous case. This allows the classic algorithm to perform better than the proposed algorithm in a narrow region around its minimum. Nevertheless in the whole input region, the proposed algorithm performs quite better. The errors are now greater since a third-order Volterra series cannot approximate well the nonlinearity $g(x)$ of infinite order. The truncation error causes the RRMSE curves of the classical algorithm rise more quickly away from the minimum.

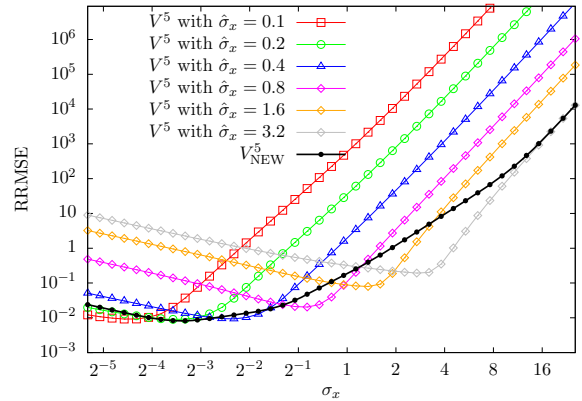


Fig. 11 Relative root mean square error of the output of fifth-order Volterra series versus the desired output of an ∞ order test Volterra system

So the region with acceptable values are narrower near their minimum for the classical algorithm. The same thing does not happen for the new algorithm, as can be clearly seen in Figs. 9, 10 and 11.

Figure 10 shows the error of fourth-order Volterra models, while Fig. 11 report result for the fifth order. The errors decrease when higher-order models are used. Also in these Figures, we can see that the Volterra series identified with the classic algorithm behave well only with inputs whose variance is near to that used in the identification, while the proposed algorithm approaches the envelope of the minimum errors.

The envelope of the minima is the ideal goal for the algorithm, and the Fig. 11 shows that for Volterra series of higher order, the proposed algorithm approaches such a goal.

5 Conclusion

An improvement in the Lee–Schetzen method is proposed. Each Wiener kernel is identified with a different variance. The new formulas needed for the Wiener to Volterra transformation are presented. The proposed method allows us to obtain acceptable errors for a wider range of input variances. The comparison with the classical method shows a reduction for both the main source of error in the identification of Volterra series: input non-ideality and order truncation. The code used to implement this new algorithm is available at [11] as MATLAB[®] compatible source code.

Acknowledgments I would like to thank Massimiliano Pirani and Claudio Turchetti for the useful discussions in the early stage of the work.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

1. Boyd, S., Chua, L.O.: Fading memory and the problem of approximating nonlinear operators with Volterra series. *IEEE Trans. Circuits Syst.* **CAS-32**(11), 1150–1161 (1985)
2. Cheng, C., Peng, Z., Zhang, W., Meng, G.: Wavelet basis expansion-based volterra kernel function identification through multilevel excitations. *Nonlinear Dyn.* **76**(2), 985–999 (2014). doi:[10.1007/s11071-013-1182-3](https://doi.org/10.1007/s11071-013-1182-3)
3. Diniz, P.S.R.: *Adaptive Filtering*. Springer, US (2008)
4. Fréchet, M.: Sur les fonctionnelles continues. *Annales de LEcole Normale Supérieure* **27**(3), 193–216 (1910)
5. Goussard, Y., Krenz, W.C., Stark, L.: An improvement of the Lee and Schetzen cross-correlation method. *IEEE Trans. Autom. Contr.* **AC-30**(9), 895–898 (1985)
6. Korenberg, M.J., Bruder, S.B., McIlroy, P.J.: Exact orthogonal kernel estimation from finite data records: extending Wiener's identification of nonlinear systems. *Ann. Biomed. Eng.* **16**, 201–214 (1988)
7. Koukoulas, P., Kalouptsidis, N.: Nonlinear system identification using gaussian inputs. *IEEE Trans. Signal Process.* **43**(8), 1831–1841 (1995)
8. Lee, Y.W., Schetzen, M.: Measurement of the Wiener kernels of a nonlinear system by crosscorrelation. *Int. J. Control* **2**(3), 237–254 (1965)
9. Marmarelis, V.Z.: *Nonlinear Dynamic Modeling of Physiological Systems*. **10**, Wiley (2004)
10. Mathews, V.J., Sicuranza, G.L.: *Polynomial Signal Processing*. Wiley, New York (2000)
11. Orcioni, S.: Volterra2.0 Identification Tool Web Site. <http://sourceforge.net/p/volterra20>
12. Orcioni, S., Pirani, M., Turchetti, C.: Advances in Lee-Schetzen method for Volterra filter identification. *Multidimens. Syst. Signal Process.* **16**(3), 265–284 (2005)
13. Orcioni, S., Pirani, M., Turchetti, C., Conti, M.: Practical notes on two Volterra filter identification direct methods. In: *Proceedings the of IEEE International Symposium on Circuits and Systems. ISCAS'02*, vol. 3, pp. 587–590. Scottsdale, Arizona (2002)
14. Pirani, M., Orcioni, S., Turchetti, C.: Diagonal kernel point estimation of n-th order discrete Volterra-Wiener systems. *EURASIP J. Appl. Signal Process.* **2004**(12), 1807–1816 (2004)
15. Sandberg, I.W.: \mathbb{R}_+ fading memory and extensions of input-output maps. *IEEE Trans. Circuits Syst.* **I49**(11), 1586–1592 (2002)
16. Schetzen, M.: A theory of non-linear system identification. *Int. J. Control* **20**(4), 577–592 (1974)
17. Silva, W.: Identification of nonlinear aeroelastic systems based on the Volterra theory: progress and opportunities. *Nonlinear Dyn.* **39**, 25–62 (2005)
18. Volterra, V.: *Theory of Functionals and of Integrals and Integro-differential Equations*. Dover Publications, New York (1959)
19. Wiener, N.: *Nonlinear Problems in Random Theory*. Wiley, New York (1958)