New Method for Identifying Finite Degree Volterra Series ⋆

Wael SULEIMAN and André MONIN

LAAS - CNRS, University of Toulouse
7 avenue du Colonel Roche, 31077 Toulouse cedex 4
France

Abstract

In this paper, the identification of a class of nonlinear systems which admits input-output maps described by a finite degree Volterra series is considered. In actual fact, it appears that this class can model many important nonlinear multivariable processes not only in engineering, but also in biology, socio-economics, and ecology.

To solve this identification problem, we propose a method based on a local gradient search in a local parameterization of the state space realization of finite degree Volterra series with infinite horizon. Using the local parameterization not only reduces the amount of the gradient calculations to the minimal value, but also overcomes the nonuniqueness problem of the optimal solution.

Moreover, we propose a sequential projection method to provide an initial estimation of the parameters of finite degree Volterra series realization. This estimation is used to initialize the gradient search method.

Key words: Identification; nonlinear systems; Volterra series; optimization.

1 Introduction

During the last thirty years, the development of identification theory of dynamic systems has been a subject of active research, such a development is the result of the necessity of reliable models in process control, aerospace applications, biomedical systems, ecology, physiology, biology, and socio-economics (Lakshmikantham, 1988; Nijmeijer & van der Schaft, 1996; Sastry, 1999; Schetzen, 1981; Korenberg & Hunter, 1990; Schuppen & Jan, 2004).

The developments of linear subspace identification methods have recently offered a significantly practical tool to deal with the identification of multivariable linear or pseudo linear system (Söderström & Stoica, 1989; Moonen *et al.*, 1989; Overschee & Moor, 1994; Viberg, 1995; Ljung, 1999).
However, in practice, the identification of nonlinear multivariable systems is becoming crucial since many systems in nature are nonlinear. In the theory of nonlinear systems, the term "nonlinear" defines a class of systems for which the linear approximation fails to be an efficient model able to capture the dynamic of the system. The class of nonlinear systems is a complex class, and defining an universal represented model of this class is a very complicated task.
Therefore, anyone interested in nonlinear modeling is compelled to focus on specific model classes (Billings, 1980; Haber & Unbeheauen, 1990; Lee, 1998; Mathews & Sicuranza, 2000; Pearson, 1995, 2000; Doyle *et al.*, 2001).

In this paper, we focus on the representation of nonlinear system by Volterra series with infinite horizon (Volterra, 1930; Brockett, 1972, 1976). The expansion of a finite degree Volterra series has the following form

$$y_t = \sum_{n=1}^{l} \sum_{\tau_1=0}^{\infty} \sum_{\tau_2=0}^{\tau_1} \ldots \sum_{\tau_n=0}^{\tau_{n-1}} w_n(\tau_1, \tau_2, \ldots, \tau_n) \times u_{t-\tau_1} \otimes u_{t-\tau_2} \otimes \ldots \otimes u_{t-\tau_n} + v_t \tag{1}$$

where $u_t \in \mathbb{R}^m$ is the input signals, $y_t \in \mathbb{R}^p$ is the output signals. The measurement noise v_t is assumed to be a white-noise that is independent of the input signal and with zero mean, and $\otimes$ denotes the Kronecker product. The functions $w_n(\tau_1, \tau_2, \ldots, \tau_n) \in \mathbb{R}^{p \times m^n}$ denote the Volterra kernels. As the last term in the series involves the l^{th} kernel, they will be called Volterra series of degree l with infinite horizon. This class of Volterra series is dense, in the L^2 sense, in nonlinear analytic input-output systems (Brockett, 1976).
Approximating nonlinear systems by Volterra series has been used in many research areas, such as filtering (Monin & Salut, 1996), predictive control (Allgöwer & Zheng,

⋆ This paper was not presented at any IFAC meeting. Corresponding author M. Wael. SULEIMAN. Tel. +33 56 133 7912. Fax. +33 56 133 6969.

Email addresses: `suleiman@laas.fr` (Wael SULEIMAN), `monin@laas.fr` (André MONIN).

2000), ecology (Takeuchi, 1996), biology (Thieme, 2003), and system control (Nijmeijer & van der Schaft, 1996; Doyle *et al.*, 2001).
In reality, the direct calculation of the Volterra kernels from Equation (1) is not feasible in practice. This because it involves the integral past of input signal and the size of the matrices which should be manipulated increases exponentially.
For that, the horizon of finite degree Volterra series is fixed (Rugh, 1981; Schetzen, 1989). The obtained class is called truncated finite degree Volterra series of degree l.
It has the following form

$$\begin{aligned} y_t = & \sum_{n=1}^{l} \sum_{\tau_1=0}^{T} \sum_{\tau_2=0}^{\tau_1} \dots \sum_{\tau_n=0}^{\tau_{n-1}} w_n(\tau_1, \tau_2, \dots, \tau_n) \\ & \times u_{t-\tau_1} \otimes u_{t-\tau_2} \otimes \dots \otimes u_{t-\tau_n} + v_t \end{aligned} \quad (2)$$

where T is the length of the considered horizon. In this case, the estimation of Volterra kernels becomes trivial and can be obtained by least-square methods. However, if we consider that the Volterra kernels are fully parameterized, then the number of parameters which should be estimated is the following [1]

$$C(l,T) = p \left\{ \sum_{n=1}^{l} \frac{(T+n)!}{n!\,T!} m^n \right\} \quad (3)$$

recall that p and m are the dimensions of the output and input signals respectively. It is clear that $C(l,T)$ increases exponentially with respect to l and T. For that, in the case of multivariable systems, we are obligated to operate high dimensional matrices, which are often ill-conditioned (Nowak & Veen, 1994).

The main contribution of this paper is defining a new method to identify a state-space realization of finite degree Volterra series with infinite horizon (1). The initial estimation of the realization's parameters is obtained by a sequential projection method that we have developed thanks to the recursive property of the realization structure, then the realization parameters are optimized using a local gradient search method.

The paper is organized as follows. In Section 2, a realization of finite degree Volterra series in finite dimension state space representation is defined. In Section 3, the output error identification problem is formulated and the structure's parameters are chosen. A local parameterization of the realization of finite degree Volterra series is developed in Section 4. In Section 5, we propose a sequential projection method to calculate an initial estimation of the structure's parameters. In Section 6, we summarize the algorithm of identification. Finally, Section 7 presents some illustrative examples and a comparison with some system identification methods for nonlinear systems.

[1] See Appendix.A for the details of calculation.

1.1 General notations

As a general rule in this paper, selecting elements of matrices is done using MATLAB™ standard matrix operations, e.g. $M(:, i:j)$ stands for the sub-matrix of the matrix M which contains the columns from the i^{th} to j^{th} columns.

2 Realization of a finite degree Volterra series

(Brockett, 1976) has proved that any observable realization of a finite Volterra series of degree l can be approximated by an recursive realization of the form

$$\begin{aligned} Z_t^1 &= A^1 Z_{t-1}^1 + B^1 u_t &&, \; Z_0^1 = 0 \\ Z_t^2 &= A^2 Z_{t-1}^2 + B^2 (u_t \otimes Z_t^1) &&, \; Z_0^2 = 0 \\ &\vdots \\ Z_t^l &= A^l Z_{t-1}^l + B^l (u_t \otimes Z_t^{l-1}) &&, \; Z_0^l = 0 \\ y_t &= \sum_{i=1}^{l} C^i Z_t^i + v_t \end{aligned} \quad (4)$$

The direct relation between y_t and u_t, can be obtained by using the property $FG \otimes HJ = (F \otimes H)(G \otimes J)$ and a simple development of (4) leads to

$$\begin{aligned} y_t = & \sum_{i=1}^{l} \sum_{\tau_1=0}^{t-1} \sum_{\tau_2=0}^{\tau_1} \dots \sum_{\tau_i=0}^{\tau_{i-1}} C^i \Phi^i(\tau_1, \dots, \tau_i) \\ & \times u_{t-\tau_1} \otimes u_{t-\tau_2} \dots \otimes u_{t-\tau_i} + v_t \end{aligned} \quad (5)$$

where

$$\begin{aligned} & \Phi^1(\tau_1) = (A^1)^{\tau_1} B^1 \\ & \text{and for i} \geq 2: \\ & \Phi^i(\tau_1, \tau_2, \dots, \tau_i) = (A^i)^{\tau_1} B^i \left[I_m \otimes \Phi^{i-1}(\tau_2, \dots, \tau_i) \right] \end{aligned} \quad (6)$$

By comparing (1) and (5), we observe that Realization (4) leads to Volterra series similar to (1), in which we suppose that $u_t = 0$ for $t \leq 0$, and the Volterra kernels are approximated by a sum of matrix products.
In fact, this approximation depends on the dimensions of the states $\{Z_t^i : i = 1, 2, \cdots, l\}$, and it is clear that it becomes better if we increase the dimensions of states. On the other hand, for practical purposes these dimensions should be chosen as lower as possible in order to find a compromise between the approximation and numerical complexity issues.
Obviously, Realization (4) is not unique, assume that the states of Structure (4) transform into $X_t^i = \left(T^i\right)^{-1} Z_t^i$ where $T^i \in \mathbb{R}^{n_i \times n_i}$ nonsingular matrices, and n_i is the dimension of

the state Z_t^i. The structure becomes

$$\begin{aligned} X_t^1 &= \bar{A}^1 X_{t-1}^1 + \bar{B}^1 u_t \\ X_t^2 &= \bar{A}^2 X_{t-1}^2 + \bar{B}^2 (u_t \otimes X_t^1) \\ &\vdots \\ X_t^l &= \bar{A}^l X_{t-1}^l + \bar{B}^l (u_t \otimes X_t^{l-1}) \\ y_t &= \sum_{i=1}^{l} \bar{C}^i X_t^i + v_t \end{aligned}$$

where

$$\begin{bmatrix} \bar{A}^i & \bar{B}^i \\ \bar{C}^i & 0 \end{bmatrix} = \begin{bmatrix} \left(T^i\right)^{-1} A^i T^i & \left(T^i\right)^{-1} B^i T_m^{i-1} \\ C^i T^i & 0 \end{bmatrix} \tag{7}$$

and $\{T_m^j : j = 0, 1, 2, \cdots, l-1\}$ are defined as follows

$$T_m^j = I_m \otimes T^j, \quad T_m^0 = I_m : I_m \text{ is the identity matrix}$$

Lemma 1 *The realization of finite degree Volterra series (4) is asymptotically stable if and only if:*

$$For\ i = 1, 2, \cdots, l : \rho(A^i) < 1 \tag{8}$$

where $\rho(A^i)$ stands for the spectral radius of A^i.

PROOF. Each subsystem may be viewed as a linear system with input $(u_t \otimes Z_t^{i-1})$. So, Z_t^i will be asymptotically stable if and only if the linear system is stable $\big(\rho(A^i) < 1\big)$ and $u_t \otimes Z_t^{i-1}$ is bounded. Assuming that Z_t^1 is stable $\big(\rho(A^1) < 1\big)$, so $u_t \otimes Z_t^1$ is bounded and Z_t^2 is stable if and only if $\big(\rho(A^2) < 1\big)$. And so on. ■

3 Output error identification

Our goal is to determine the coefficient matrices of Structure (4). Assume that all matrices are fully parameterized, so the structure's parameters can be given by

$$\theta = \begin{bmatrix} vec(A^1) \\ vec(B^1) \\ vec(C^1) \\ \vdots \\ vec(A^l) \\ vec(B^l) \\ vec(C^l) \end{bmatrix} \tag{9}$$

where vec(.) denotes the vectorization operator defined as follows

$$\begin{aligned} & vec : M \in \mathbb{R}_{m\times n} \to \mathbb{R}_{m\cdot n} \\ & vec(M) = vec\Big[m_1\ \ m_2 \cdots m_n\Big] = \Big[m_1^T\ \ m_2^T \cdots m_n^T\Big]^T \end{aligned}$$

Given the input u_t and output y_t of the real system, the model corresponding to θ can be given as follows

$$\begin{aligned} \hat{Z}_t^1 &= A^1(\theta)\hat{Z}_{t-1}^1 + B^1(\theta) u_t \\ \hat{Z}_t^2 &= A^2(\theta)\hat{Z}_{t-1}^2 + B^2(\theta)(u_t \otimes \hat{Z}_t^1) \\ &\vdots \\ \hat{Z}_t^l &= A^l(\theta)\hat{Z}_{t-1}^l + B^l(\theta)(u_t \otimes \hat{Z}_t^{l-1}) \\ \hat{y}_t(\theta) &= \sum_{i=1}^{l} C^i(\theta)\hat{Z}_t^i \end{aligned} \tag{10}$$

Note that as Z_t^i depends on θ, the mapping $\hat{y}_t(\theta)$ is nonlinear with θ. Our goal is achieved if the output $\hat{y}_t(\theta)$ approximates the output of the real system accurately. This criterion can be transformed into the minimization of the output error with respect to the parameters θ. Considering a data of length N, the output-error cost function is given by

$$J_N(\theta) = \frac{1}{N}\sum_{k=1}^{N} \|y_k - \hat{y}_k(\theta)\|_2^2 = \frac{1}{N} E_N(\theta)^T E_N(\theta) \tag{11}$$

where

$$E_N(\theta) = \big[e(1)^T\ e(2)^T \cdots e(N)^T\big]^T \tag{12}$$

is the error vector in which $e(k) = y_k - \hat{y}_k(\theta)$. The minimization of (11) is clearly a nonlinear, nonconvex optimization problem. The numerical solution of this problem can be calculated by different algorithms, e.g. gradient search method (Levenberg-Marquard method) is a popular one. This iterative method is based on the updating of the system parameters θ as follows

$$\theta^{i+1} = \theta^i - (\psi_N^T(\theta^i)\psi_N(\theta^i) + \lambda^{i+1} I)^{-1} \psi_N^T(\theta^i) E_N(\theta^i) \tag{13}$$

Where λ^i is the regularization parameter and

$$\psi_N(\theta) \triangleq \frac{\partial E_N(\theta)}{\partial \theta^T} \tag{14}$$

is the jacobian of the error vector $E_N(\theta)$. As we mentioned in Section (2), the structure (4) is not unique. As a consequence, the minimization of $J_N(\theta)$ does not have a unique solution. The nonuniqueness solution of θ is the consequence of the full parameterization of the matrices of the realization.
However, the optimal solution can be made unique by choosing a suitable canonical parameterization. As each subsystem of the realization can be considered as a linear system, one could use a classical parameterization of the various parameterization proposed for the linear systems. Unfortunately, these parameterizations are not numerically robust

(McKelvey & Helmersson, 1997).
To overcome the nonuniqueness problem of the optimal θ and keep the full parameterization of the matrices, Ribarits *et al* (Ribarits *et al.*, 2004) have proposed a method for the linear systems, in which the directions that do not change the cost of output error function are identified and projected out at each iteration.
For that only the active parameters are updated. In analogues way, we will define a local parameterization of the realization of finite degree Volterra series (4) in order to define the directions in which the cost function $J_N(\theta)$ does not change.

4 Local parameterization

Recall that two realizations of a finite degree Volterra series (4) are similar if their coefficient matrices are related by Equation (7), where the transformation matrices $\{T^i : i = 1,2,\cdots,l\}$ parameterize the subset of equivalent models. The obtained subset defines a manifold. In order to identify the tangent plane of this manifold at $(A^i, B^i, C^i : i = 1,2,\cdots,l)$, we linearize Relation (7) around the identity matrices. Considering a small perturbation $T^i = I_{n_i} + \Delta T^i$, we suppose that $\rho(\Delta T^i) \ll 1$ then by using the approximation $\left(I_{n_i} + \Delta T^i\right)^{-1} \simeq I_{n_i} - \Delta T^i$ and neglecting all second order terms, we obtain

$$\begin{bmatrix} \bar{A}^i & \bar{B}^i \\ \bar{C}^i & 0 \end{bmatrix} = \begin{bmatrix} A^i & B^i \\ C^i & 0 \end{bmatrix} + \begin{bmatrix} -\Delta T^i A^i + A^i \Delta T^i & -\Delta T^i B^i \\ C^i \Delta T^i & 0 \end{bmatrix} + \begin{bmatrix} 0 & B^i \Delta T_m^{i-1} \\ 0 & 0 \end{bmatrix} \tag{15}$$

By defining

$$\Lambda_i^j \triangleq \left[\, 0_{n_j \times (i-1)n_j} \;\; I_{n_j} \;\; 0_{n_j \times (m-i)n_j} \,\right]$$

it is possible to write ΔT_m^j as follows

$$\Delta T_m^j = \sum_{i=1}^{m} \Lambda_i^{j^T} \Delta T^j \Lambda_i^j$$

If we consider the following vectors of parameters

$$\theta = \begin{bmatrix} vec(A^1) \\ vec(B^1) \\ vec(C^1) \\ \vdots \\ vec(A^l) \\ vec(B^l) \\ vec(C^l) \end{bmatrix} \quad and \quad \bar{\theta} = \begin{bmatrix} vec(\bar{A}^1) \\ vec(\bar{B}^1) \\ vec(\bar{C}^1) \\ \vdots \\ vec(\bar{A}^l) \\ vec(\bar{B}^l) \\ vec(\bar{C}^l) \end{bmatrix} \tag{16}$$

the relation between θ and $\bar{\theta}$ can be obtained using the property $vec(ABC) = \left(C^T \otimes A\right) vec(B)$

$$\bar{\theta} = \theta + M_\theta \begin{bmatrix} vec(\Delta T^1) \\ \vdots \\ vec(\Delta T^l) \end{bmatrix} \tag{17}$$

where for $1 \leq j \leq l-1$

$$M_\theta\left(:, 1 + \sum_{k=1}^{j-1} n_k^2 : \sum_{k=1}^{j} n_k^2\right) = \begin{bmatrix} \mathbf{0}_{\alpha_j \times n_j^2} \\ -\left(A^j\right)^T \otimes I_{n_j} + I_{n_j} \otimes A^j \\ -\left(B^j\right)^T \otimes I_{n_j} \\ I_{n_j} \otimes C^j \\ \mathbf{0}_{n_{j+1}^2 \times n_j^2} \\ \sum_{i=1}^{m} \Lambda_i^{j^T} \otimes \left(B^{j+1} \Lambda_i^{j^T}\right) \\ \mathbf{0}_{\gamma_j \times n_j^2} \end{bmatrix}$$

and

$$M_\theta\left(:, 1 + \sum_{k=1}^{l-1} n_k^2 : \sum_{k=1}^{l} n_k^2\right) = \begin{bmatrix} \mathbf{0}_{\alpha_l \times n_l^2} \\ -\left(A^l\right)^T \otimes I_{n_l} + I_{n_l} \otimes A^l \\ -\left(B^l\right)^T \otimes I_{n_l} \\ I_{n_l} \otimes C^l \end{bmatrix} \tag{18}$$

with

$$\begin{aligned} \alpha_j &= \sum_{i=1}^{j-1} n_i(n_i + m n_{i-1} + p) \\ \gamma_j &= p n_{j+1} + \sum_{i=j+2}^{l} n_i(n_i + m n_{i-1} + p) \end{aligned}$$

Lemma 2 *The left null space of M_θ (18) defines a basis of the directions in which the parameters should be modified to lead a change in the value of cost function $J_N(\theta)$.*

PROOF. Equation (17) shows that, the tangent space of the manifold of equivalent realizations at $(A^i, B^i, C^i : i = 1,2,\cdots,l)$ is equal to the column space of the matrix M_θ (18). Since the left null space of the matrix M_θ is orthogonal complement to the column space, the directions in which the value of the cost function $J_N(\theta)$ changes are those related to left null space of M_θ. ■

Let the QR decomposition of M_θ be given by

$$M_\theta = \left[\, Q_1 \;\; Q_2 \,\right] \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \tag{19}$$

then a basis of the null space of M_θ is Q_2.
Thus, the update rule should be modified such that we project out the directions in which the cost function does not change. The new update rule becomes

$$\theta^i = \theta^{i-1} - Q_2\left(Q_2^T\psi_N^T\psi_N Q_2 + \lambda^i I\right)^{-1} Q_2^T\psi_N^T E_N \tag{20}$$

where Q_2, ψ_N and E_N depend on θ^{i-1}. Note that since Q_2 depends on the past parameter θ^{i-1}, the QR decomposition (19) must be computed at each iteration.

4.1 Computing the iterative parameter update

In order to compute the update rule (20), the quantities $E_N(\theta)$ and $\psi_N(\theta)$ should be computed. Computing the vector $E_N(\theta)$ can be done by simulating the system (10) with $\theta = \theta^{i-1}$. At the same time, this simulation brings out the states $\{\hat{Z}_t^i : i = 1, 2, \cdots, l\}$. In order to simulate the gradient $\psi_N(\theta^{i-1})$, we should compute the derivative of $\hat{y}_t$ with respect to θ^{i-1}. Let us define

$$\zeta_{t,k}^j = \frac{\partial \hat{Z}_t^j}{\partial \theta_k} \tag{21}$$

then the computation of $\frac{\partial \hat{y}_t}{\partial \theta^T} = \left[\frac{\partial \hat{y}_t}{\partial \theta_1} \ \cdots \ \frac{\partial \hat{y}_t}{\partial \theta_q}\right]$, where q is the number of parameters in θ, can be made as follows

$$\begin{aligned}
\zeta_{t,k}^1 &= A^1\zeta_{t-1,k}^1 + \frac{\partial A^1}{\partial\theta_k}\hat{Z}_{t-1}^1 + \frac{\partial B^1}{\partial\theta_k}u_t\\
\zeta_{t,k}^2 &= A^2\zeta_{t-1,k}^2 + \frac{\partial A^2}{\partial\theta_k}\hat{Z}_{t-1}^2 + \frac{\partial B^2}{\partial\theta_k}(u_t\otimes\hat{Z}_t^1) + B^2(u_t\otimes\zeta_{t,k}^1)\\
&\vdots\\
\zeta_{t,k}^l &= A^l\zeta_{t-1,k}^l + \frac{\partial A^l}{\partial\theta_k}\hat{Z}_{t-1}^l + \frac{\partial B^l}{\partial\theta_k}(u_t\otimes\hat{Z}_t^{l-1}) + B^l(u_t\otimes\zeta_{t,k}^{l-1})\\
\frac{\partial\hat{y}_t}{\partial\theta_k} &= \sum_{j=1}^{l} C^j\zeta_{t,k}^j + \frac{\partial C^j}{\partial\theta_k}\hat{Z}_t^j
\end{aligned} \tag{22}$$

In fact, if we consider the local parameterization of finite degree Volterra realization, then computing the update rule (20) can be done without calculate ψ_N first. Let us define

$$\Omega_N \triangleq \psi_N Q_2 = \frac{\partial E_N}{\partial\theta^T}Q_2 \tag{23}$$

So, the update rule can be expressed as follows

$$\theta^i = \theta^{i-1} - Q_2\left(\Omega_N^T\Omega_N + \lambda^i I\right)^{-1}\Omega_N^T E_N \tag{24}$$

Consider the tp elements of the s^{th} column of Ω_N

$$\Omega_N((t-1)p+1 : tp, s) = \sum_{k=1}^{q}\frac{\partial\hat{y}_t}{\partial\theta_k}Q_2(k,s) \tag{25}$$

In order to calculate this sum, let us define

$$\zeta_t^j = \sum_{k=1}^{q}\frac{\partial\hat{Z}_t^j}{\partial\theta_k}Q_2(k,s) \tag{26}$$

Using (22), the computation of (25) can be done as follows

$$\begin{aligned}
\zeta_t^1 &= A^1\zeta_{t-1}^1 + \Delta A^1\hat{Z}_{t-1}^1 + \Delta B^1 u_t\\
\zeta_t^2 &= A^2\zeta_{t-1}^2 + \Delta A^2\hat{Z}_{t-1}^2 + \Delta B^2(u_t\otimes\hat{Z}_t^1) + B^2(u_t\otimes\zeta_t^1)\\
&\vdots\\
\zeta_t^l &= A^l\zeta_{t-1}^l + \Delta A^l\hat{Z}_{t-1}^l + \Delta B^l(u_t\otimes\hat{Z}_t^{l-1}) + B^l(u_t\otimes\zeta_t^{l-1})\\
\sum_{k=1}^{q}\frac{\partial\hat{y}_t}{\partial\theta_k}Q_2(k,s) &= \sum_{j=1}^{l} C^j\zeta_t^j + \Delta C^j\hat{Z}_t^j
\end{aligned} \tag{27}$$

where

$$\Delta A^j = \sum_{k=1}^{q}\frac{\partial A^j}{\partial\theta_k}Q_2(k,s) \quad : j = 1, 2, \cdots, l \tag{28}$$

and $\Delta B^j, \Delta C^j$ are defined analogously. These matrices can be obtained from

$$\begin{bmatrix} vec(\Delta A^1)\\ vec(\Delta B^1)\\ vec(\Delta C^1)\\ \vdots\\ vec(\Delta A^l)\\ vec(\Delta B^l)\\ vec(\Delta C^l)\end{bmatrix} = \sum_{k=1}^{q}\frac{\partial\theta}{\partial\theta_k}Q_2(k,s) = \sum_{k=1}^{q} e^k Q_2(k,s) \tag{29}$$

where

$$e^k = [0 \ \dots \ 0 \ \underset{\substack{\uparrow\\ k}}{1} \ 0 \ \dots \ 0]^T$$

Note that the number of columns of Ω_N is smaller than that of ψ_N because the first one is the number of active parameters and the second one is the total number of parameters.

5 Computing an initial estimation

The reason why a good initial estimation is crucial is that the local gradient search method converges to the nearest local optimum in the neighborhood of initial guess of solution. Moreover, the initial estimation should also provide a stable realization that verifies the constrain (8).

In this section, we propose a sequential projection method in order to provide an initial estimation of the realization's vector of parameters (θ). The basic idea is to use the consecutive property of finite degree Volterra series realization to define a sequential projection procedure. Recall that, the error introduced in the projection process is due to the non orthogonality of the subprocesses (linear, quadratic, cubic, $\cdots$), and as well the effect of the noise on the observed output signal. The algorithm of projection can be resumed as follows

(1) Estimate the best linear stable approximation (the linear subsystem) of the nonlinear system.
(2) From the simulation of the linear subsystem, we project the residual on the class of 2-degree subsystems.
(3) From the simulation of linear + 2-degree subsystems, we project the residual on the class of 3-degree subsystem.
(4) Etc.

Moreover, this projection procedure yields an estimation of the dimensions of the states $\{Z_t^i : i = 1,2,\cdots,l\}$, which means the set $\{n_i : i = 1,2,\cdots,l\}$. Furthermore, it yields the degree of the realization (l) by defining a precision criterion as it will appear in the sequel.

5.1 Identification of linear subsystem

Estimating the best linear stable approximation of nonlinear system can be formulated as a minimization problem

$$\begin{aligned} &min \;\; L_N^1 = \frac{1}{N}\sum_{t=1}^{N} \parallel y_t - y_t^1 \parallel_2^2 \\ &\text{where} \\ &Z_t^1 = A^1 Z_{t-1}^1 + B^1 u_t \\ &y_t^1 = C^1 Z_t^1 \\ &\text{with the constraint} \\ &\rho\left(A^1\right) < 1 \end{aligned} \tag{30}$$

The parameters of the optimization problem, which should be estimated, are the triple (A^1, B^1, C^1). The above minimization problem can be transformed into the following matricial form (Gopinath, 1969; DeMoor *et al.*, 1988)

$$\begin{aligned} &min \;\; \parallel Y_{\alpha,N} - \Gamma_\alpha^1 Z_{0,N-\alpha}^1 - \Phi_\alpha^1 U_{\alpha,N} \parallel_F^2 \\ &\text{subject to} \\ &\rho\left(A^1\right) < 1 \end{aligned} \tag{31}$$

where $\parallel . \parallel_F$ denotes Frobenius norm. The input and output are stocked in Hankel matrices form

$$U_{\alpha,N} \triangleq \begin{pmatrix} u_1 & u_2 & \cdots & u_{N-\alpha+1} \\ u_2 & u_3 & \cdots & u_{N-\alpha+2} \\ \vdots & \vdots & \ddots & \vdots \\ u_\alpha & u_{\alpha+1} & \cdots & u_N \end{pmatrix}$$

where α and N refer to the number of rows in the matrix and data length respectively. The output Hankel matrix $Y_{\alpha,N}$ is defined analogously to $U_{\alpha,N}$. The matrices Γ_α^1, Φ_α^1 and $Z_{0,N-\alpha}^1$ are defined as follows

$$\Gamma_\alpha^1 = \begin{bmatrix} C^1 \\ C^1 A^1 \\ \vdots \\ C^1 (A^1)^{(\alpha-1)} \end{bmatrix}$$

$$\Phi_\alpha^1 = \begin{bmatrix} C^1 B^1 & 0 & 0 & \cdots & 0 \\ C^1 A^1 B^1 & C^1 B^1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ C^1 (A^1)^{(\alpha-1)} B^1 & \cdots & \cdots & \cdots & C^1 B^1 \end{bmatrix}$$

$$Z_{0,N-\alpha}^1 = \left[Z_0^1 \; Z_1^1 \; \cdots Z_{N-\alpha}^1 \right].$$

Note that the number of rows (α) should be roughly chosen to be greater than the expected linear system order n_1. This condition guarantees that the extended matrix of observability Γ_α^1 has a full rank. Our objective is to estimate the matrices Γ_α^1 and Φ_α^1. Since only the data matrices $U_{\alpha,N}$ and $Y_{\alpha,N}$ are known, instead of solving the minimization problem (30) with respect to Γ_α^1 and Φ_α^1, we can transform it into an equivalent one involving only the matrix Γ_α^1 as follows

$$\begin{aligned} &min \;\; \parallel Y_{\alpha,N} \Pi^\perp_{U^T_{\alpha,N}} - \Gamma_\alpha^1 Z_{0,N-\alpha}^1 \Pi^\perp_{U^T_{\alpha,N}} \parallel_F^2 \\ &\text{subject to} \\ &\rho\left(A^1\right) < 1 \end{aligned} \tag{32}$$

where $\Pi^\perp_{U^T_{\alpha,N}}$ is the orthogonal projection onto the nullspace of $U_{\alpha,N}$

$$\Pi^\perp_{U^T_{\alpha,N}} = I - U^T_{\alpha,N}\left(U_{\alpha,N} U^T_{\alpha,N}\right)^{-1} U_{\alpha,N} \tag{33}$$

such that

$$U_{\alpha,N} \Pi^\perp_{U^T_{\alpha,N}} = \mathbf{0} \tag{34}$$

The inverse of the matrix $\left(U_{\alpha,N} U^T_{\alpha,N}\right)$ exists if the input is persistently exciting and $N > m\alpha$. In fact, the origin of the idea of subtracting the term that involves the input signal

belongs to the direct 4SID method, which is a classical subspace method (DeMoor *et al.*, 1988).
To solve the problem (32), one could use the Singular Value Decomposition (SVD). Recall that, the extended observability matrix (Γ_α^1) is a full rank matrix. Consider the following SVD

$$Y_{\alpha,N}\Pi^{\perp}_{U^T_{\alpha,N}} = \begin{bmatrix} \mathscr{Q}_s & \mathscr{Q}_n \end{bmatrix} \begin{bmatrix} \mathscr{S}_s & 0 \\ 0 & \mathscr{S}_n \end{bmatrix} \begin{bmatrix} \mathscr{V}_s^T \\ \mathscr{V}_n^T \end{bmatrix} \tag{35}$$

where the matrix $\mathscr{S}_s$ contains the principals singular values (further than threshold).
The dimension of this matrix yields n_1 (the dimension of Z_t^1).
The estimation of the matrix Γ_α^1 can be done by solving the following problem of minimization

$$\begin{aligned} & min \quad \| \Gamma_\alpha^1 - \mathscr{Q}_s \|_F^2 \\ & \text{subject to} \\ & \rho\left(A^1\right) < 1 \end{aligned} \tag{36}$$

By using the property

$$\Gamma_\alpha^1(1:(\alpha-1)p,:)A^1 = \Gamma_\alpha^1(p+1:\alpha p,:) \tag{37}$$

and replacing the stability condition $\rho(A^1) < 1$ by equivalent Lyapunov inequalities

$$\rho(A^1) < 1 \iff \exists P \geq \delta I_{n_1} : \; P - A^1 P {A^1}^T \geq \delta I_{n_1}$$

where $\delta > 0$, we can transform the minimization problem (36) into the following one

$$\begin{aligned} & min\ J\left(A^1\right) \triangleq \| L_{A^1}\left(\mathscr{Q}_s^{\dagger} - \mathscr{Q}_s^{\ddagger} A^1\right) R_{A^1} \|_F^2 \\ & \text{subject to} \\ & \begin{bmatrix} P - \delta I_{n_1} & X^T \\ X & P \end{bmatrix} \geq 0_{2n_1} \end{aligned} \tag{38}$$

where

$$\begin{aligned} \mathscr{Q}_s^{\dagger} &\triangleq \mathscr{Q}_s(p+1:\alpha p,:) \\ \mathscr{Q}_s^{\ddagger} &\triangleq \mathscr{Q}_s(1:(\alpha-1)\,p,:) \\ X &\triangleq A^1 P \end{aligned} \tag{39}$$

If we let $R_{A^1} \triangleq P$ and L_{A^1} be equal to the identity matrix, the problem is converted to an optimization problem that involves minimizing a linear function over symmetric cones and can be solved using SeDuMi MATLAB™ package (Sturm, 1999) (see (Lacy & Bernstein, 2003) for more details).

Solving the minimization problem (38) provides an estimation of a stable matrix A^1. The matrix C^1 can be estimated using the following formula

$$C^1 = \left(\sum_{k=0}^{\alpha-1} \mathscr{Q}_s\left(kp+1:(k+1)p,:\right) \right) \Lambda_{A^1}^T \left(\Lambda_{A^1}\Lambda_{A^1}^T\right)^{-1} \tag{40}$$

where $\Lambda_{A^1} \triangleq I_{n_1} + \sum_{i=1}^{\alpha-1}\left(A^1\right)^i$. Note that as A^1 verifies that $\rho(A) < 1$, Λ_{A^1} is a full rank matrix and the matrix inverse exists. The matrix Φ_α^1 can be estimated by considering the least-squares solution to the overdetermined system of equations

$$\mathscr{Q}_n^T Y_{\alpha,N} U_{\alpha,N}^T \left(U_{\alpha,N} U_{\alpha,N}^T\right)^{-1} = \mathscr{Q}_n^T \Phi_\alpha^1 \tag{41}$$

Finally, the matrix B^1 can be easily calculated from the estimation of Φ_α^1.

5.2 *Identification of higher order subsystems*

After using the procedure described in the previous section, an estimation of (A^1, B^1, C^1) is available. Calculating the state Z_t^1 and the output y_t^1 for $t = 1,..,N$ of the linear subsystem can be done by evaluating the following model

$$\begin{aligned} Z_t^1 &= A^1 Z_{t-1}^1 + B^1 u_t \\ y_t^1 &= C^1 Z_t^1 \end{aligned}$$

The next task is to project the residual on the class of 2-degree subsystem. We define the residual as follows

$$\tilde{y}_t = y_t - y_t^1 \tag{42}$$

The estimation of the best 2-degree subsystem can be done by solving the following optimization problem

$$\begin{aligned} & min \quad L_N^2 = \frac{1}{N}\sum_{t=1}^{N} \| \tilde{y}_t - y_t^2 \|_2^2 \\ & \text{where} \\ & Z_t^2 = A^2 Z_{t-1}^2 + B^2\left(u_t \otimes Z_t^1\right) \\ & y_t^2 = C^2 Z_t^2 \\ & \text{with the constraint} \\ & \rho\left(A^2\right) < 1 \end{aligned} \tag{43}$$

As $u_t \otimes Z_t^1$ is computed thanks to the previous step, this minimization problem is similar to (30). So, by using an analogous logic, we obtain an estimation of the matrices A^2, B^2, C^2.

Then, this procedure is reiterated until we get an appreciated precision. Defining this precision criterion is the objective of the following section.

5.3 Determining the realization's degree

Consider that, the subsystems of order inferior to j are available. In order to verify that the nonlinear system is accurately fitted by the finite degree Volterra series realization, we evaluate the following criterion

$$\hat{y}_t = \sum_{i=1}^{j} y_t^i$$
$$\left(1 - \frac{\sum_{t=1}^{N} (y_t - \hat{y}_t)^T (y_t - \hat{y}_t)}{\sum_{t=1}^{N} (y_t - \bar{y})^T (y_t - \bar{y})}\right) \times 100 \geq \eta \quad (44)$$

where η is a user-defined constant and $\bar{y}$ denotes the mean of the output signals defined as follows

$$\bar{y} = \frac{1}{N} \sum_{t=1}^{N} y_t \quad (45)$$

If the criterion (44) is true, the algorithm stops and the realization's degree is j. Otherwise we estimate the subsystem of order $j+1$.
Note that the measurement noise v_t should be considered when we define η as follows

$$0 < \eta < \left(1 - \frac{\sum_{t=1}^{N} v_t^T v_t}{\sum_{t=1}^{N} (y_t - \bar{y})^T (y_t - \bar{y})}\right) \times 100 \quad (46)$$

6 Identification algorithm

We can resume the algorithm of identification as follows

(1) Calculate an initial estimation of the realization's parameters using the sequential projection described in Section 5. Recall that, this procedure yields the realization degree (l), the dimensions of the states Z_t^i $\{n_i : i = 1,2,\cdots,l\}$, and the matrices $\{A^i, B^i, C^i : i = 1,2,\cdots,l\}$. The estimated vector of parameters θ^0 is used as an initial guess for the optimization process and $k \leftarrow 0$.
(2) Calculate the states $\{Z_t^i : i = 1,2,\cdots,l\}$ and $\hat{y}_t$ by simulating the system (10) with $\theta = \theta^k$.
(3) Compute $E_N(\theta)$ using (12).
(4) Calculate the matrix M_θ using (18).
(5) Calculate the QR decomposition of M_θ (19), from which we obtain Q_2.
(6) Calculate $\Delta A^j, \Delta B^j$ and ΔC^j $: j = 1,2,\cdots,l$ using (29).
(7) Calculate the matrix Ω_N using (25) and (27), we suppose that $\{\zeta_0^j = \mathbf{0} \ : j = 1,2,\cdots,l\}$.
(8) Calculate the update rule of the gradient search algorithm using (24) and $k \leftarrow k+1$.
(9) Perform the termination test for minimization, If true, the algorithm stops. Otherwise, return to step (2), i.e. compute the values $J_N(\theta^{k-1})$ and $J_N(\theta^k)$ using (11) and test if $\|J_N(\theta^k) - J_N(\theta^{k-1})\|_2$ is small enough.

7 Illustrative examples

In this section, first we compare the computational complexity of the direct gradient search and gradient search in the local parameterization of finite degree Volterra realization as a function of realization degree. At the same time, we corroborate that the method is able to identify the realization of finite degree Volterra series. Second, we consider two examples of nonlinear system, and we compare the capability of finite degree Volterra realization to approximate the considered examples with some of state-of-the-art system identification methods for nonlinear systems.
In order to validate the obtained models, for a data of N samples, we have used $T_{id} = \frac{N}{2}$ samples for the identification purpose, and the rest of them $T_{val} = T - T_{id} = \frac{N}{2}$ samples for the validation purpose. The validation step, can be viewed as an evaluation of prediction accuracy of the models, and in order to verify that the models are capable to extract the real output signal from the measured noised one, we consider the output signal *without* the measurement noise in the validation step. The model accuracy is defined as the Percent Variance Accounted For $(\%VAF)$

$$\%VAF \triangleq \left(1 - \frac{\sum_{t=1}^{N} (y_t - \hat{y}_t)^T (y_t - \hat{y}_t)}{\sum_{t=1}^{N} (y_t - \bar{y})^T (y_t - \bar{y})}\right) \times 100$$

where $\hat{y}_t$ denotes the estimated output signal and $\bar{y}$ is the mean of the output signals.

7.1 Computational complexity

In this section, we figure out that using a local parameterization accelerates the convergence of gradient search method. For that, we consider a finite degree Volterra series realization (4), where the states $\{Z_t^i : \ i = 1,2,\cdots,l\}$ have the same dimension which is equal to $n = 10$. The identification experiment for each value of the realization's degree is repeated 10 times. Each one has a length of 4000 samples. The average of optimization time is then computed.
The optimization time is the required time to reach a specific precision (local optimum). Each experiment has three inputs ($u_t \in \mathbb{R}^3$) which have been chosen to be uniform white noise and three outputs ($y_t \in \mathbb{R}^3$) . The measurement noise v_t is a Gaussian white noise scaled such that the signal to noise ratio $SNR = 10\ dB$. The initial estimation of the realization's vector of parameters (θ) is computed using the sequential projection method explained in Section 5. The computation time of sequential projection method is calculated and given in Fig. 2. The optimization time as a function of realization degree (l) for the direct gradient search method and the gradient search in local parameterization space is given in

Fig. 1. The implementation of two approaches is done using MATLAB™ programming environment.

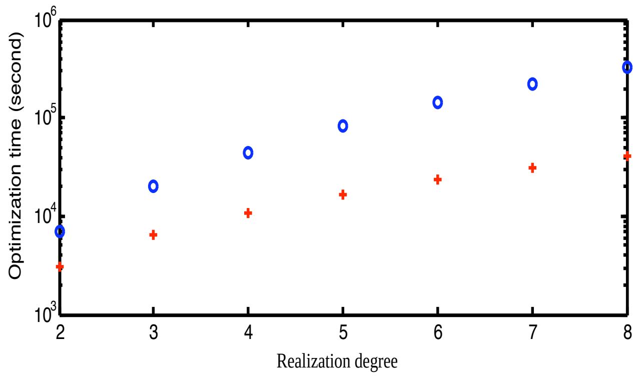


Fig. 1. Optimization time for the direct gradient search method (circle mark) and gradient search in local parameterization space (plus mark).

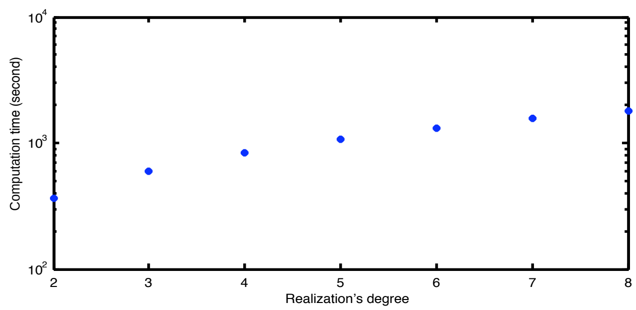


Fig. 2. Computation time of the sequential projection method.

It is clear from Fig. 1 that using local parameterization reduces the time of computation. This reducing becomes advantageous when the realization degree increases.

Table 1
Accuracy of the identified models

Identification	Initial estimated model	Optimized model
Accuracy $(\%VAF)$	82.3 ± 2.7	89.1 ± 1.8
Validation		
Accuracy $(\%VAF)$	86.9 ± 1.3	98.6 ± 0.7

In order to corroborate that the proposed method is able to identify the finite Volterra series efficiently, we calculate the average and the standard deviation of the accuracies of the obtained models. The accuracies are reported in Table 1 where the initial estimated model is the model obtained by the sequential projection explained in Section 5 and the optimized model is that obtained by local gradient search in local parameterization space. These accuracies demonstrate that the proposed method has efficiently identified the finite degree Volterra realization and the sequential projection method provides a good initial guess for the optimization process.

7.2 First example of nonlinear system

Consider the following example

$$\begin{aligned} x_t^1 &= 0.7\, x_{t-1}^1 + u_t \qquad &, x_0^1 = 0 \\ x_t^2 &= 0.7\, x_{t-1}^2 + \left(x_t^1\right)^2 + x_t^1 + u_t \qquad &, x_0^2 = 0 \\ y_t &= \left(x_t^1\right)^2 + x_t^2 + v_t \end{aligned} \tag{47}$$

The input signal $u_t \in \mathbb{R}$ is chosen to be uniform white noise with standard deviation is equal to 0.5 and zero mean, its length is equal to 1000 samples. The measurement noise v_t is a Gaussian white noise scaled such that, the signal to noise ratio $SNR = 10\ dB$. We compare the performance of finite Volterra series realization with neural network based nonlinear system identification, which is a popular approach (Billings *et al.*, 1992; Sjoberg *et al.*, 1994; Nørgaard *et al.*, 2000). For comparison, we have chosen two nonlinear model structures

(1) Neural Network Output Error (NNOE) model with 10 hidden hyperbolic tangent units in the hidden layer. The output function generated by the neural network can be calculated as follows

$$\begin{aligned} \varphi_t &= \left[\hat{y}_{t-1}(\theta)\ \hat{y}_{t-2}(\theta)\ \hat{y}_{t-3}(\theta)\ u_t\ u_{t-1}\ u_{t-2}\right]^T \\ \hat{y}_t(\theta) &= g\left(\varphi_t, \theta\right) \end{aligned} \tag{48}$$

where φ_t is a vector containing the regressors, θ is a vector containing the weights and g is the function realized by the neural network.

(2) Neural Network State Space Innovation Form (NNSSIF) model which determines a nonlinear state space model of the dynamic system

$$\begin{aligned} \hat{x}_t(\theta) &= g\left(\hat{x}_{t-1}(\theta), u_t\right) \\ \hat{y}_t &= C\hat{x}_t(\theta) \end{aligned} \tag{49}$$

where $\hat{x}_t(\theta)$ is the state vector and its dimension is n_x, g is the function realized by the neural network and C is a constant matrix.
In this example, the best performance was obtained by $n_x = 6$ and neural network with 10 hidden hyperbolic tangent units in the hidden layer. The neural network is trained with the Levenberg-Marquardt method.

The implementation of two models (NNOE and NNSSIF) is done using Neural Network Based System Identification toolbox (Nørgaard, 2000).

Note that the dimensions of the states $(Z_t^i :\ i = 1, 2)$ of the quadratic system (Volterra series realization of degree two)

Table 2
Accuracy of the identified models

Identification	Quadratic system	NNOE model	NNSSIF model
Accuracy (%VAF)	85.7	76.5	63.9
Validation			
Accuracy (%VAF)	98.0	81.9	73.4

are 2 and 3 respectively. Recall that these dimensions are obtained using the sequential projection method explained in Section 5. The accuracies reported in Table 2 show the outperformance of a quadratic system in comparison with NNOE and NNSSIF models.

To deal with a clear graphical representation, we represent the results in the validation interval over a window of 150 samples in Fig. 3.

7.3 Second example of nonlinear system

Consider the following example of nonlinear system

$$
\begin{aligned}
x_t^1 =&0.8x_{t-1}^1 + 0.8u_t(1) + 0.6u_t(2) + 0.8u_t(3)\\
x_t^2 =&0.5\left(x_{t-1}^1\right)^2 + 0.5x_{t-1}^2 + 0.9\left(u_t(2)\right)^3 + u_t(1)u_t(2)x_{t-1}^1\\
x_t^3 =&0.7x_{t-1}^2 + 0.7x_{t-1}^3 + 0.8u_t(2)u_t(3)x_{t-1}^1 + 0.9\left(u_t(1)\right)^3\cdots\\
&+0.9\left(u_t(2)\right)^3\\
y_t =& \begin{pmatrix} 0.6 & 0.3 & 0.6\\ 0.2 & 0.7 & 0.6\\ 0.3 & 0.0 & 0.5\end{pmatrix} \begin{pmatrix} x_t^1\\ x_t^2\\ x_t^3\end{pmatrix} + v_t
\end{aligned}
\tag{50}
$$

where the non observed nonlinear state is $x_t = \left[x_t^1,\ x_t^2,\ x_t^3\right]^T$, and $u_t(i)$ denotes the i^{th} element of the input signal $u_t \in \mathbb{R}^3$. The system is simulated with three dimensional uniform white noise with standard deviation is equal to 1 as input and with the initial condition $\left[x_0^1,\ x_0^2,\ x_0^3\right]^T = \left[0,\ 0,\ 0\right]^T$. The length of the input data is equal to 2000 samples. The measurement noise v_t is a Gaussian white noise is scaled such that the $SNR = 15\,dB$. We compare the performance of finite degree Volterra series realization with NNSSIF model which can handle Multiple Inputs - Multiple Outputs (MIMO) systems. The best performance was obtained by $n_x = 10$ and 15 hidden hyperbolic tangent units in the hidden layer.

Note that the dimensions of the states ($Z_t^i:\ i = 1,2,3$) of the cubic system (Volterra series realization of degree three) are 6, 9 and 13 respectively. The accuracies reported in Table 3 show the outperformance of a cubic system in comparison with NNSSIF model. To deal with a clear graphical representation, we represent the results in the validation interval over a window of 150 samples in Fig. 4.

Table 3
Accuracy of the identified models

Identification	Accuracy ($\%VAF$)	
	Cubic system	NNSSIF model
First output	96.8	79.2
Second output	96.9	82.4
Third output	96.0	86.5
Validation		
First output	98.0	82.3
Second output	97.6	83.8
Third output	97.3	88.5

It well be of interest to compare the performance of finite Volterra series realization with truncated Volterra series which has the same degree. Therefore, the accuracies and the number of parameters of truncated Volterra series of degree three with various horizon lengths T are calculated and reported in Table 4. Note that the accuracy reported in Table 4 is the total accuracy of the output signal $y_t \in \mathbb{R}^3$.

Table 4
Accuracy comparison between cubic system and truncated Volterra series of degree three

	Number of parameters	Identification Accuracy	Validation Accuracy
Cubic system	901	96.7	97.9
	Truncated Volterra series of degree three and horizon *T*		
$T=7$	10764	99.1	38.4
$T=9$	19395	99.2	59.9
$T=10$	25047	99.4	61.6
$T=11$	31698	99.3	61.4
$T=12$	39429	99.0	58.8
$T=15$	69912	99.4	48.4

From Table 4, three remarks can be concluded

(1) The number of parameters of cubic system is much smaller than that of truncated Volterra series of degree three .
(2) The validation accuracy of cubic system is much higher than that of truncated Volterra series of degree three. This result is particularly interesting. The reason why the prediction of cubic system is more accurate than truncated Volterra series is that the cubic system consider the integral past of the input signal and its parameters are optimized such that the output of the model accurately approximates the real output of system. On the contrary, the truncated Volterra series consider a fixed horizon and calculate the Volterra kernels which correspond to this horizon.

(3) On the contrary of theoretical expectations, the validation accuracy of truncated Volterra series decreases with increasing the horizon T. This because the number of parameters increases and as well the size of matrices which become ill-conditioned.

8 Conclusion

In this paper, we have presented a new method to identify a recursive state-space realization of finite degree Volterra series with infinite horizon. The method is based on a local parameterization of the state-space representation of the realization and subsequent gradient search in the resulting local parameter space. Furthermore, we have proposed a sequential projection procedure to calculate an initial estimation of the realization's parameters.

The method has successfully applied to identify various illustrative examples, and a comparison with some methods from the state-of-the-art nonlinear system identification methods have pointed out the outperformance of finite Volterra series realization.
Moreover, a comparison with truncated Volterra series of the same degree has borne out that not only the number of parameters of finite Volterra series realization is much smaller than that of truncated Volterra series, but also the prediction accuracy of finite Volterra series realization is superior to that of truncated Volterra series.

Our future work will focus on the improvement of the optimization algorithm in order to guarantee the convergence to the global optimum. Moreover, we will work on the development of control techniques for this realization in order to control real world nonlinear systems.

A Appendix

Consider that the total number of parameters of a truncated Volterra series of degree l and horizon length T is $C(l,T)$. Let us define $C_n(T)$ as the total number of parameters of the Volterra kernels of degree n (the number of elements of the matrices $w_n(\tau_1,\tau_2,\ldots,\tau_n),\ 0\le\tau_n\le\tau_{n-1}\le\cdots\le\tau_1\le T$). It is obvious that

$$C(l,T)=\sum_{n=1}^{l}C_n(T) \tag{A.1}$$

Suppose S is any finite set of elements and denote the number of these elements $\sharp S$, then

$$\begin{aligned} C_n(T) &= \sharp\{0\le\tau_n\le\tau_{n-1}\le\cdots\le\tau_1\le T\}\,pm^n \\ &= \sum_{k=0}^{T}\sharp\{0\le\tau_n\le\tau_{n-1}\le\cdots\le\tau_1\le k\}\,pm^n \\ &= \sum_{k=0}^{T}\binom{k+n-1}{n-1}pm^n \end{aligned} \tag{A.2}$$

where $\binom{\cdot}{\cdot}$ denotes the binomial coefficient defined as follows

$$\binom{N}{K}=\frac{N!}{K!\,(N-K)!} \tag{A.3}$$

Using the following properties

$$\begin{aligned} \binom{K}{N} &= \binom{N}{N-K} \\ \textstyle\sum_{k=0}^{T}\binom{k+q}{k} &= \binom{T+q+1}{T} \end{aligned} \tag{A.4}$$

One could rewrite $C_n(T)$ as follows

$$C_n(T)=\binom{T+n}{T}pm^n=\binom{T+n}{n}pm^n \tag{A.5}$$

Finally, by using (A.1) and (A.5) we obtain

$$\begin{aligned} C(l,T) &= p\left\{\sum_{n=1}^{l}\binom{T+n}{n}m^n\right\} \\ &= p\left\{\sum_{n=1}^{l}\frac{(T+n)!}{n!\,T!}m^n\right\} \end{aligned} \tag{A.6}$$

References

Allgöwer, F. & Zheng, A. (2000). *Nonlinear Model Predictive Control*. Birkhäuser.

Billings, S. (1980). Identification of nonlinear systems: a suvey. *IEE Proceedings*, **127**, 272–285.

Billings, S., Jamaluddin, H. & Chen, S. (1992). Properties of neural networks with applications to modelling nonlinear dynamical systems. *Int. J. Control*, **55**, 193–224.

Brockett, R. (1972). System theory on group manifolds and coset spaces. *SIAM J.Control*, **10**, 265–284.

Brockett, R. (1976). Volterra series and geometric control theory. *Automatica*, **12**, 167–176.

DeMoor, B., Vandewalle, J., Vandenberghe, L. & Mieghem, P.V. (1988). A geometrical strategy for the identication of state space models of linear multivariable systems a geometrical strategy for the identication of state space models of linear multivariable system with singular value decomposition. In *Proc. IFAC 88*, 700–704, Beijing, China.

Doyle, F.J., Pearson, R.K. & Ogunnaike, B.A. (2001). *Identification and Control Using Volterra Models*. Springer.

Gopinath, B. (1969). On the identification of linear time-invariant systems from input-ouput data. *Bell Syst. Tech. J.*, **48**, 1101–1113.

Haber, R. & Unbeheauen, H. (1990). Structure identification of nonlinear dynamic systems - a survey on input/output approaches. *Automatica*, **26**, 651–677.

Korenberg, M. & Hunter, I. (1990). The identification of nonlinear biological systems: Wiener kernel approaches. *Ann Biomed Eng.*, **18**, 629–654.

Lacy, S.L. & Bernstein, D.S. (2003). Subspace identification with guaranteed stability using constrained optimization. *IEEE Trans. Automatic Control*, **48**, 1259–1263.

Lakshmikantham, V. (1988). *Stability Analysis of Nonlinear Systems*. Marcel Dekker.

Lee, J. (1998). Modeling and identification for nonlinear predictive control: Requirements, current status and future research needs. In *International Symposium on Nonlinear Model Predictive Control: Assessment and Future Directions*, 91–107, Ascona, Switzerland.

Ljung, L. (1999). *System Identification: Theory for the User*. Prentice Hall Informations and Systems Sciences, 2nd edn.

Mathews, V. & Sicuranza, G. (2000). *Polynomial Signal Processing*. Wiley.

McKelvey, T. & Helmersson, A. (1997). System identification using an over-parametrized model class- improving the optimization algorithm. In *31th IEEE Conf. Decision Control*, San Diego, California USA.

Monin, A. & Salut, G. (1996). I.I.R. Volterra filtering with application to bilinear systems. *IEEE Trans. Signal Processing*, **44**, 2209–2221.

Moonen, M., Moor, B., Vandenberghe, L. & Vandewalle, J. (1989). On- and off-line identification of linear state-space models. *Int. J. Control*, **49**, 219–232.

Nijmeijer, H. & van der Schaft, A. (1996). *Nonlinear Dynamical Control Systems*. Springer.

Nørgaard, M. (2000). *Neural Network Based System Identification Toolbox*. Department of Automation, Department of Mathematical Modeling, Technical University of Denmark.

Nørgaard, M., Ravn, O., Poulsen, N.K. & Hansen, L.K. (2000). *Neural networks for Modelling and Control of Dynamic Systems*. Springer-Verlag, London, UK.

Nowak, R. & Veen, B.V. (1994). Efficient methods for identification of Volterra filter models. *Signal Processing*, **38**, 417–428.

Overschee, P.V. & Moor, B.D. (1994). N4SID: subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, **30**, 75–93.

Pearson, R. (1995). Nonlinear input/output modeling. *J. Process Control*, **5**, 197–211.

Pearson, R. (2000). *Discrete-Time Dynamic Models*. Oxford.

Ribarits, T., Deistler, M. & McKelvey, T. (2004). An analysis of the parametrization by data driven local coordinates for multivariable linear systems. *Automatica*, **40**, 789–803.

Rugh, W.J. (1981). *Nonlinear System Theory: The Volterra/Wiener Approach*. The Johns Hopkins.

Sastry, S. (1999). *Nonlinear Systems: Analysis, Stability and Control*. Springer.

Schetzen (1989). *The Volterra and Wiener theories of nonlinear systems*. New York: Wiley-Interscience.

Schetzen, M. (1981). Nonlinear system modeling based on the wiener theory. *Proc. of IEEE*, **69**, 1557–1573.

Schuppen, V. & Jan, H. (2004). System theory for system identification. *J. Econometrics*, **118**, 313–339.

Sjoberg, J., Hjalmarsson, H. & Ljung, L. (1994). Neural networks in system identification. In *10th IFAC symposium on system identification*, Copenhagen, Denmark.

Söderström, T. & Stoica, P. (1989). *System identification*. Prentice Hall, New York.

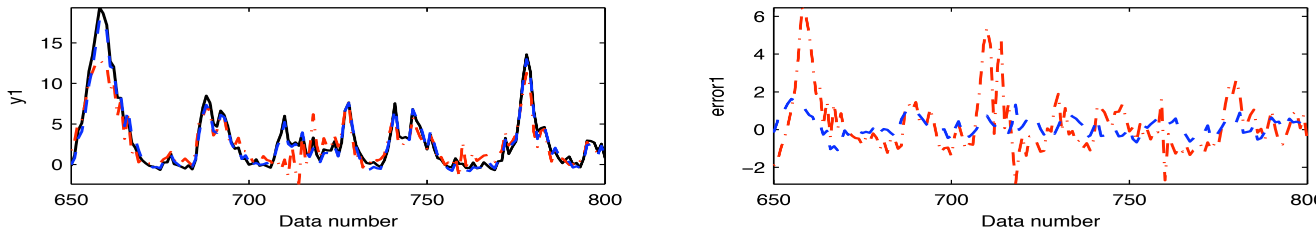
Sturm, J. (1999). *Using SeDuMi 1.0x, a MATLAB Toolbox for Optimization Over Symmetric Cones*. Dept. Quantiative Economics, Maastricht Univ, Netherlands.

Takeuchi, Y. (1996). *Global Dynamical Properties of Lotka-Volterra Systems*. World Scientific.

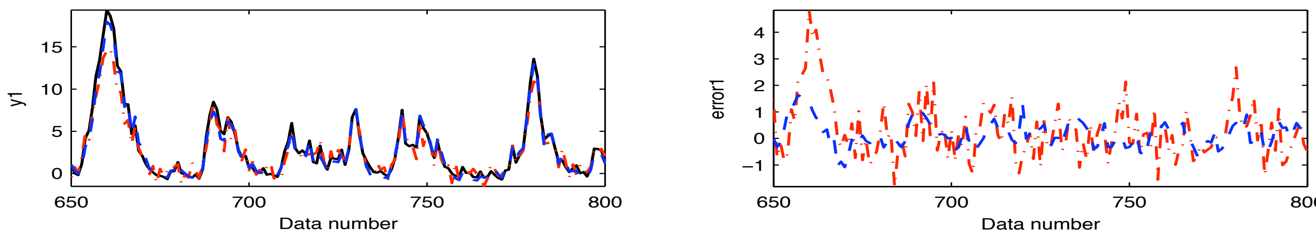
Thieme, H.R. (2003). *Mathematics in Population Biology*. Princeton University Press.

Viberg, M. (1995). Subspace-based methods for the identification of linear time-invariant systems. *Automatica*, **31**, 1835–1851.

Volterra, V. (1930). *Theory of Functionals and of Integral and Integro-Differential Equations*. Dover Publications.



(a) The real output *without* the measurement noise (solid line) with the outputs of quadratic system (dashed line) and NNOE model (dash-dotted line), and their errors.



(b) The real output *without* the measurement noise (solid line) with the outputs of quadratic system (dashed line) and NNSSIF model (dash-dotted line), and their errors.

Fig. 3. Comparison of the results of quadratic system, NNOE model, and NNSSIF model.

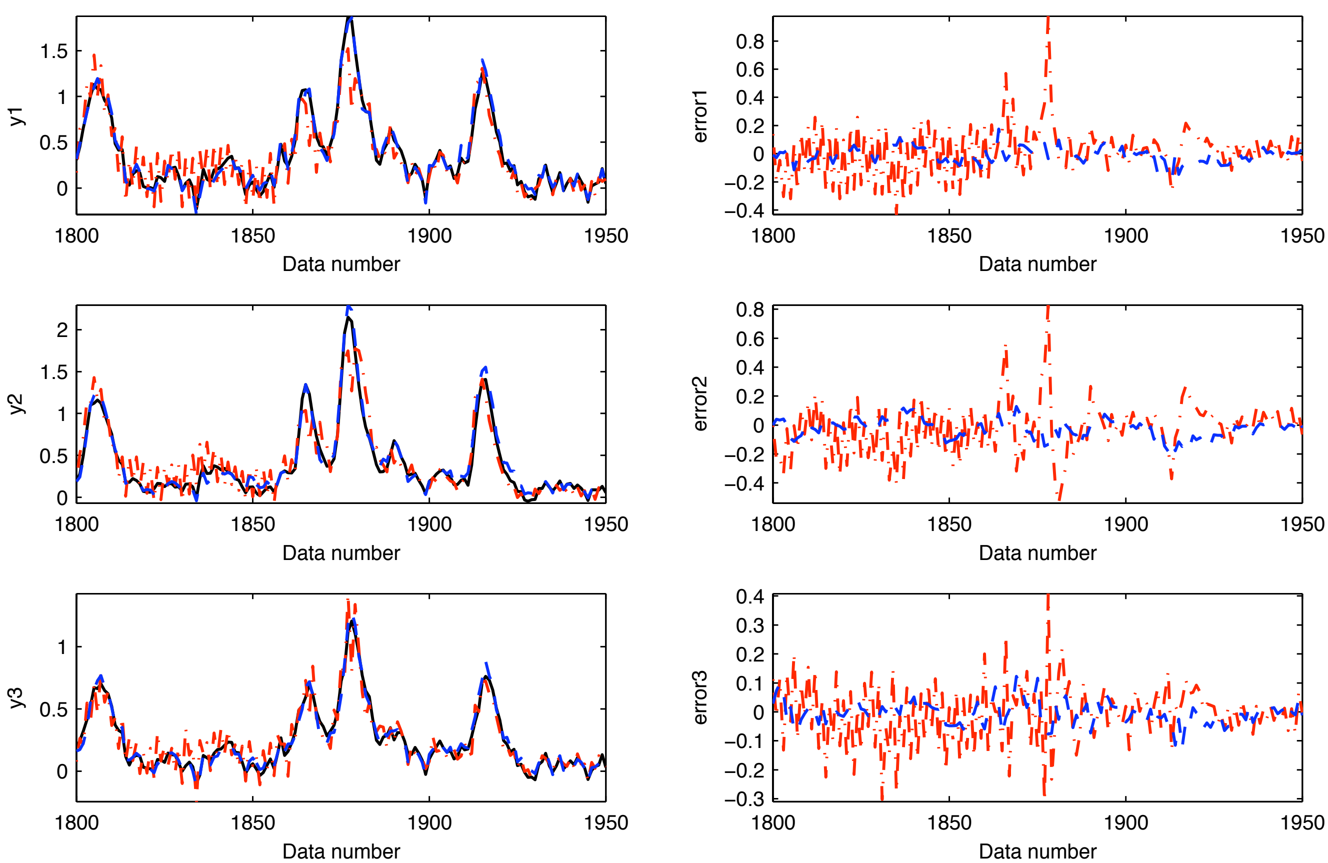


Fig. 4. The real outputs *without* the measurement noise (solid line) with the outputs of cubic system (dashed line) and NNSSIF model (dash-dotted line), and their errors are superimposed.