# A REGULARIZATION METHOD FOR NONLINEAR INVERSE PROBLEMS BY USING A VOLTERRA MODEL

**G. Bibes*†, R. Ouvrard*, P. Coirault***

* Laboratoire d'Automatique et d'Informatique Industrielle, LAII-ESIP
40 Avenue du recteur Pineau, 86022 Poitiers Cedex, France regis.ouvrard@esip.univ-poitiers.fr
† Partially supported by Région Poitou-Charentes

**Keywords:** Ill-posed inverse problem, nonlinear system, regularization, Volterra model

## Abstract

In this paper, a nonlinear inverse problem is proposed. The main idea is to apply the well known Tikhonov regularization to a Volterra model. The Volterra model allows to represent a large variety of nonlinearities, but for inverse problem, a nonlinear optimization procedure must be used and an important computational burden is required. A particular computational implementation is proposed to reduce the computational burden and to simplify the use. Here the Volterra model consists in an approximation of each Volterra kernel by a multidimensional orthonormal basis expansion.

## 1 Introduction

In many fields of science one is interested in inverse problems. These ones allow to reconstruct a quantity which is not directly accessible by using an experimentally measurable quantity. They are generally known to be ill-posed problems. The solution is to use a prior knowledge in a criterion optimization approach to change the ill-posed problem into a better-conditioned problem. The most popular method is the Tikhonov regularization method [**?**].
In this paper we propose a regularization method for nonlinear systems by using a black-box continuous-time model. This latter consists in modelling each kernel of truncated Volterra series of the nonlinear system by a multidimensional orthonormal basis.
This paper is organized as follows. The next section presents the modelling used in this approach. In the section 3, a regularization method with a particular formulation is described. Simulation results are presented in section 4 and concluding remarks in section 5.

## 2 Volterra series modelling

### 2.1 Volterra series

Volterra series allow to represent the input-output behaviour of nonlinear systems. There is a large literature on Volterra series, e.g. [**?**, **?**, **?**, **?**]. Here, we are considered the continuous-time case, but the regularization described below can also be applied to the discrete-time case.

It is well known that under certain conditions the relationship between the output $y(t)$ and the input $u(t)$ of a nonlinear time invariant system can be written in the Volterra series form :

$$\begin{array}{l} y\left(t\right) = y_0 + \int_0^t h_1\left(\tau_1\right) u\left(t-\tau_1\right) d\tau_1 \\ + \int_0^t \int_0^t h_2\left(\tau_1,\tau_2\right) \times \\ u\left(t-\tau_1\right) u\left(t-\tau_2\right) d\tau_1 d\tau_2 + \cdots \\ + \int_0^t \cdots \int_0^t h_n\left(\tau_1,\cdots,\tau_n\right) \times \\ u\left(t-\tau_1\right) \cdots u\left(t-\tau_n\right) d\tau_1 \cdots d\tau_n + \cdots \end{array} \quad (1)$$

where $h_n\left(\tau_1,\cdots,\tau_n\right)$ is called the $n$th-order Volterra kernel. The condition $h_n\left(\tau_1,\cdots,\tau_n\right) = 0$ if $\tau_i < 0$ for $i = 0,\ldots,n$ is required in order to assure the causality of the system. For simplicity let us assume that $y_0 = 0$.

### 2.2 Volterra model

The Volterra series (1) has an infinite dimension. For a practical use, a truncation is necessary. The $N$th-order truncated Volterra series can be written as

$$\begin{array}{l} \mathrm{y}\left(\mathrm{t}\right) = H_1\left[u\left(t\right)\right] + \cdots + H_n\left[u\left(t\right)\right] \\ \qquad + \cdots + H_N\left[u\left(t\right)\right] \end{array} \quad (2)$$

where

$$\begin{array}{l} H_n\left[u\left(t\right)\right] = \int_0^t \cdots \int_0^t h_n(\tau_1,\cdots,\tau_n) \\ \times\, u\left(t-\tau_1\right) \cdots u\left(t-\tau_n\right) d\tau_1 \cdots d\tau_n \end{array} \quad (3)$$

Many Volterra kernel modelling methods exist in literature. We can quote Hammerstein, Wiener or Uryson models which are subclasses of Volterra models [**?**].
The continuous-time Volterra model [**?**, **?**] considered in this paper consists of an expansion on multidimensional orthonormal basis for each kernel.
The $n$-dimensional orthonormal basis in $L^2(T^n)$, where $T$ is the set of times $T = [0,\infty[$, can be obtained from monodimensional orthonormal basis[1] $\{\phi_m(t)\}$ in $L^2(T)$ by forming the products $\{\phi_{m_1} \otimes \cdots \otimes \phi_{m_n}\}$. We denote by $\otimes$ the Hadamar product, *i.e.* given two vectors $v$, $w$, then $v \otimes w$ is the vector whose $i$th component is $v_i w_i$. The $n$-dimensional orthonormal basis is written $\{\phi_{m_1 \cdots m_n}(t)\}$.
We use the generalized basis functions $B_m(s)$ [**?**, **?**] defined by

[1] $\phi_m(t)$ is response of the ortonormal function $B_m(s)$ to the input $u(t)$

$$B_m(s) = \frac{\sqrt{2\mathrm{Re}\{a_m\}}}{s+a_m} \prod_{k=1}^{m-1} \frac{s-\overline{a_k}}{s+a_k} \tag{4}$$

where $\overline{a_k}$ denotes complex conjugate of $a_k$. They allow to introduce a variety of poles.
Notice that orthonormal basis functions can be different for each kernel in order to fit at best nonlinear dynamics.
Each kernel $H_n[u(t)]$ with $n = 1, \cdots, N$ is modelled by

$$\begin{aligned} \mathrm{H_n}[u(t), \theta_n] = \sum_{m_1=1}^{M_n} \cdots \sum_{m_n=1}^{m_{n-1}} b_{m_1 \cdots m_n} \\ \times \phi_{m_1}(t) \otimes \cdots \otimes \phi_{m_n}(t) \end{aligned} \tag{5}$$

where $\theta_n$ is the vector composed of parameters $b_{m_1 \cdots m_n}$.
Hence the truncated Volterra series (2) is approximated by

$$\begin{aligned} \mathrm{y}(\mathrm{t}, \mathrm{u}, \theta) = H_1[u(t), \theta_1] + \cdots + H_n[u(t), \theta_n] \\ + \cdots + H_N[u(t), \theta_N] \end{aligned} \tag{6}$$

with

$$\theta = [\theta_1, \cdots, \theta_N] \tag{7}$$

The Volterra model (6) is linear in parameters. Thus the parameter estimate is performed by least-squares.
The main advantage of this modelling is that no hypothesis is needed on kernel structure. The drawback is that an a priori knowledge is necessary about the dominant modes of the nonlinear system to choose the orthonormal basis function poles. In practice, to choose the number $N$ of kernels and the number $M_i$ of basis functions for each kernel, one must proceed empirically by increasing progressively $N$ and $M_i$ for $i = 1, \cdots, N$.

**Remark 1** *Notice that, for the regularization described below, the Volterra series modelling may be different from the one described in this section. The sole constraint is that from model we can compute discrete-time impulse response coefficients.*

**Remark 2** *We chose this model for several reasons. Continuous-time model has been preferred to discrete-time for both the independence from sample time choice and the relative parsimony, i.e. the relative small number of parameters. Orthonormal basis expansion can be viewed like black-box model and, despite the multidimensional basis, the model remains easily usable. Lastly, such a model allows to represent a large class of nonlinear system.*

# 3 Regularization method

## 3.1 Preliminaries

For inverse problems, the difficulty is the ill-posedness. At least one of the three necessary conditions of existence, uniqueness and stability of Hadamard is not satisfied. The solution is to use prior knowledge to change the ill-posed problem into a well-posed problem. There exists at least two general approaches that are different by the kind of considered prior information. The first approach uses a deterministic prior information and is generally called regularization method. The second uses a probabilistic prior information and we can quote Bayesian approaches or maximum entropy methods [**?**].
In this paper, a regularization method is proposed.
A general approach [**?**] to find regularized solutions $\hat{u}$ to linear inverse problem $y = Au$ is to minimize a criterion as follows

$$\hat{u} = \arg\min_u J_\lambda(u) \tag{8}$$

with

$$J_\lambda(u) = \|y^* - Au\|^2 + \lambda \|Lu\|^2, \lambda > 0 \tag{9}$$

$y^*$ denotes the finite data vector of the output measurements. The scalar $\lambda$ is the so-called regularization parameter. $L$ is an operator for which the identity or the second derivative is frequently used. The first term on the right-hand side in (9) represents the data adequation part, whereas the second term is introduced to penalize the roughness of the estimate.
The solution of (8) is

$$\hat{u} = [A'A + \lambda L'L]^{-1} A'y^* \tag{10}$$

Notice that if $\lambda = 0$ the equation (10) is the least-squares solution.

## 3.2 Problem formulation

In this paper, an extension of classical regularization method (8) to nonlinear Volterra models is provided.
In the criterion (9), the linear regression $y = Au$ is replaced with the nonlinear Volterra model output $y = F[u]$ given by (6). Then the criterion is nonlinear in parameters $u$ and a nonlinear optimization scheme is required to solve the inverse problem. Classical optimization algorithms require at least the computation of the gradient vector defined by

$$\nabla F[u] = \begin{pmatrix} \partial_1 F[u] \\ \vdots \\ \partial_{N_s} F[u] \end{pmatrix} \tag{11}$$

where $N_s$ denotes the number of samples where the input is estimated, generally the number of output measurements.
There are two main problems for the numerical implementation of nonlinear regularization. The first is to have a nonlinear model that allows to compute each component of the gradient vector. The second problem is the computational burden imposed by gradient vector simulation.

## 3.3 Computational implementation

In order to apply the Volterra model inversion to a set of experimental data, we must compute the gradient vector of the model output (6) with the derivative variable $u(t)$. Hence we are choosing to simulate each orthonormal basis function $B_m(s)$

by using the discrete convolution and the impulse response coefficients as follows

$$\Phi_m = K_m U \tag{12}$$

where $\Phi_m$ ( *res.* $U$) is a vector composed of $\phi_m(kT_s)$ ( *res.* $u(kT_s)$) with $k = 0, \cdots, N_s - 1$. $K_m$ is $N_s \times N_s$ matrix defined by

$$K_m = \begin{bmatrix} h_0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ h_1 & h_0 & \ddots & & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & & & \vdots \\ h_I & & \ddots & \ddots & \ddots & & \vdots \\ 0 & \ddots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & & \ddots & h_0 & 0 \\ 0 & \cdots & \cdots & h_I & \cdots & h_1 & h_0 \end{bmatrix} \tag{13}$$

where $h_i$ are the impulse response coefficients of the orthonormal basis function $B_m(s)$.
Then, the model of the $n$th-order Volterra kernel given by (5) can be rewritten as

$$\begin{aligned} H_n[U, \theta_n] = \sum_{m_1=1}^{M_n} \cdots \sum_{m_n=1}^{m_n-1} b_{m_1 \cdots m_n} \\ \times (K_{m_1} U) \otimes \cdots \otimes (K_{m_n} U) \end{aligned} \tag{14}$$

**Remark 3** *This formulation presents two main interests. The first is that the simulation consists only in doing matrix products. The second is the easiness to compute the gradient vector*

$$\frac{\partial \Phi_m}{\partial U} = K_m \tag{15}$$

*and*

$$\begin{aligned} \frac{\partial (\Phi_{m_i} \otimes \Phi_{m_j})}{\partial U} = \frac{\partial \Phi_{m_i}}{\partial U} \otimes \Phi_{m_j} \\ + \frac{\partial \Phi_{m_j}}{\partial U} \otimes \Phi_{m_i} \end{aligned} \tag{16}$$

### 3.4 Optimization algorithm

To perform the minimization of the criterion (9), we consider the Levenberg - Marquardt algorithm [**?**] given by

$$U_{i+1} = U_i + \Delta U \tag{17}$$

where $\Delta U$ is the solution of

$$[\mu_i + J''_\lambda(U_i)]\, \Delta U = -J'_\lambda(U_i) \tag{18}$$

The gradient $J'_\lambda(U_i)$ is computed by using (15) and (16). The hessian is given by the Gauss-Newton approximation.
Notice that unfortunately global minimization is not guaranteed but classical tools or strategies can be used to approach the optimal solution.

## 4 Simulation results

The objective of this section is to illustrate the regularization method we proposed in previous section.
The system we will study is a quadratic system given by

$$\begin{cases} \dot{x} = x + x^2 + u(t), x(0) = 0 \\ y = x \end{cases} \tag{19}$$

The simulated input range is defined by $0 \leq u \leq 1$. The sample time is $0.01s$. The data sequence used to estimate the model is not the same that the one used to apply the inverse problem.
The model is defined by (6) with $N = 2$, i.e. two Volterra kernels are considered, and each kernel is given by (5) with $M_1 = M_2 = 4$. The orthonormal function poles are the same for the two kernels $a_1 = 0.5, a_2 = 1, a_3 = 1.5, a_4 = 2$. The size of the parameter vector (7) is 14. Notice that the fact to increase the number of kernels $N$ or the number of orthonormal functions $M_i$ does not allow to obtain a better estimate.
To compute the discrete impulse response coefficients (13), we assume a zero order hold on the input and the sample time is equal to $0.01s$.
In the optimization algorithm (17), the initial value of the input vector is $U_0 = [0, \cdots, 0]$, i.e. not any a priori knowledge is used.

### 4.1 Deterministic case

In this section, a deterministic case is considered to validate both model and inversion method. For this we minimize the criterion (9) without regularization term, i.e. $\lambda = 0$. Figure 1 shows the inverse problem result. The real input is correctly represented by the input obtained by inversion.

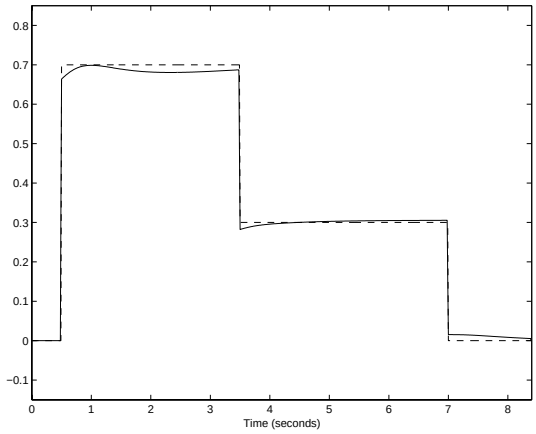


Figure 1: Inverse problem result in deterministic case. The dashed curve is real input. The solid curve is the input obtained by inversion.

### 4.2 Stochastic case

In this section, a stochastic case is considered to show the behaviour of the above method in the presence of noisy data. A white Gaussian noise such that SNR = 20dB is added to the

simulated output sequence.
The minimization of criterion (9) is performed with different regularization parameter values $\lambda$ and two different operators $L$.
In a first time, the operator $L$ is the identity, i.e. $L$ is a matrix with the coefficients

$$\begin{array}{l} L_{ij} = \delta_{ij}, i = 1, \cdots, N_l, j = 1, \cdots, N_s \\ N_l = N_s \end{array} \quad (20)$$

Figures 2 and 3 show the influence of regularization parameter $\lambda$ upon the obtained input smoothness. If $\lambda$ is too small, the regularization effect vanishes and an ill-conditioned solution is obtained. On the other hand, too large values of $\lambda$ produce smooth estimates that may be unable to represent input variation. This observation is classical in regularization and the choice of $\lambda$ is the crucial point of regularization methods.

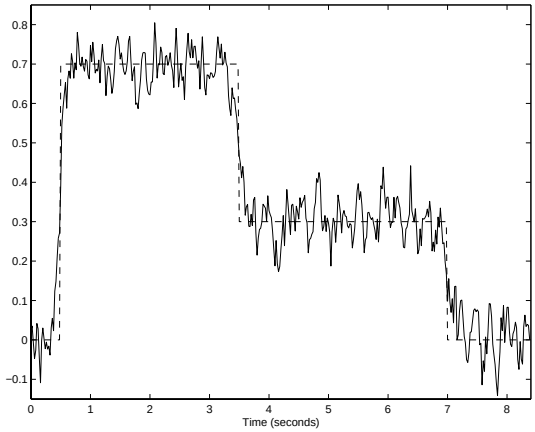


Figure 2: Identity operator $L$ and $\lambda = 0.005$. The dashed curve is real input. The solid curve is the input obtained by inversion.

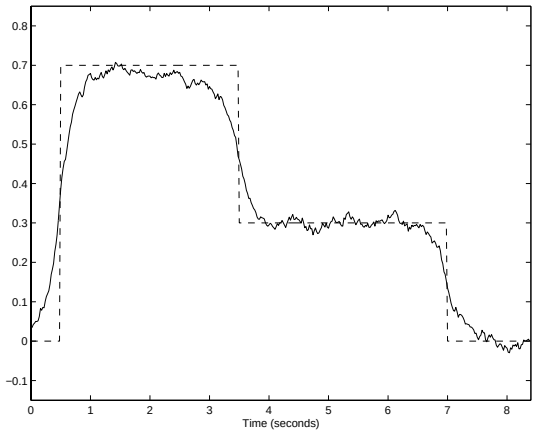


Figure 3: Identity operator $L$ and $\lambda = 0.02$. The dashed curve is real input. The solid curve is the input obtained by inversion.

In a second time, the operator $L$ is the second derivative, i.e. $L$ is a matrix with the coefficients

$$\begin{array}{l} L_{ij} = \frac{1}{T_s^2}\left(\delta_{ij} + \delta_{(i+2)j} - 2\delta_{(i+1)j}\right), \\ i = 1, \cdots, N_l, j = 1, \cdots, N_s, \\ N_l = N_s - 2. \end{array} \quad (21)$$

The results are given in figures **??** and **??**. We can see that the second derivative operator gives more smooth results.

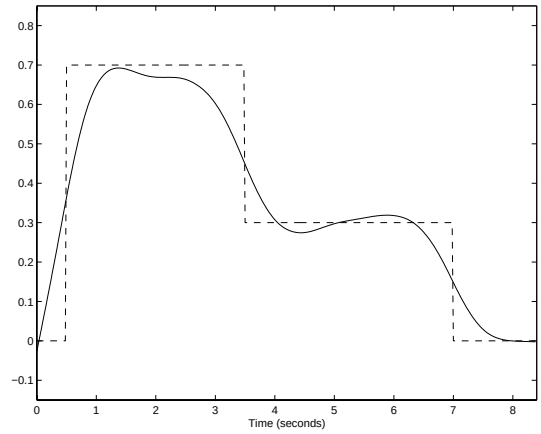


Figure 4: Second derivative operator $L$ and $\lambda = 0.005$. The dashed curve is real input. The solid curve is the input obtained by inversion.

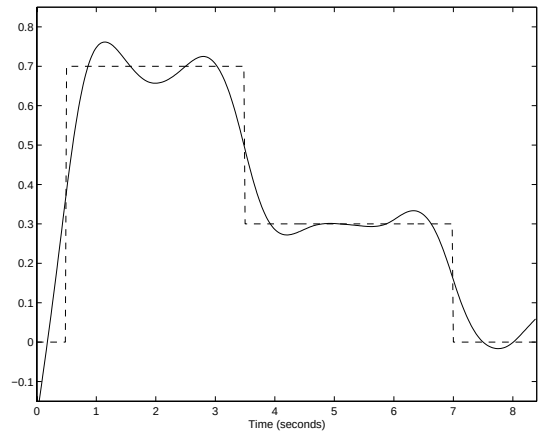


Figure 5: Second derivative operator $L$ and $\lambda = 0.001$. The dashed curve is real input. The solid curve is the input obtained by inversion.

## 5 Conclusions

A regularization method for nonlinear inverse problems has been presented. The solution is based on Tikhonov regularization, and it presents the particularity to use the well known Volterra model. The method was tested using simulated data and led to good results. The choice of the regularization parameter presents the same difficulty as in linear case.
The prospects are to optimize the regularization parameter choice, to reduce the computational burden and to extend to an iterative scheme for a real time application.

## References


[1] H. Akçay, B. Ninness. "Orthonormal basis functions for modelling continuous-time systems", *Signal Processing*, **Volume 77**, pp. 261-274, (1999).

[2] H.W. Chen. "Modeling and identification of parallel nonlinear systems : Structural classification and parameter estimation methods", *In Proc. of IEEE*, **Volume 83**, pp. 39-66, (1995).

[3] F.J. Doyle, R.K. Pearson, B.A. Ogunnaike. *"Identification and Control Using Volterra Models"*, Springer, London, (2002).

[4] S. Hassouna. "Identification des systèmes continus non linéaires par développement sur les séries de volterra", *Thèse de l'Université de Poitiers*, (2001).

[5] S. Hassouna, P. Coirault, R. Ouvrard. "Continuous nonlinear system identification using volterra series expansion", *In Proc. of American Control Conf. (Arlington, Virginia)*, pp. 4874-4879, (2001).

[6] D.W. Marquardt. "An algorithm for least-squares estimation of non-linear parameters", *J. Soc. Indust. Appl. Math.*, **Volume 11-2**, (1963).

[7] A. Mohammad-Djafari, J.-F. Giovannelli, G. Demoment, J. Idier. "Regularization, maximum entropy and porbabilistic methods in mass spectrometry data processing problems", *International Journal of Mass Spectrometry*, **Volume 215**, pp. 175-193, (2002).

[8] R. Ouvrard, P. Coirault, T. Poinot, J.C. Trigeassou. "Continuous time system identification with orthonormal functions -a new approach-", *In Proc. of European Control Conf. (Karlsruhe, Germany)*, (1999).

[9] M. Schetzen. *"The Volterra And Wiener Theories Of Nonlinear Systems"*, John Wiley and Sons, New York, NY, (1980).

[10] A.N. Tikhonov and V.Y. Arsenin. *"Solutions of Ill-posed Problems"*, Wiston, Washington, DC, (1977).

[11] Q. Zheng and E. Zafiriou. "Nonlinear system identification for control using volterra-laguerre expansion", *In Proc. of American Control Conf. (Seattle, WA)*, pp. 2195-2199, (1995).