A CONSTRAINED OPTIMISATION APPROACH TO THE BLIND ESTIMATION OF VOLTERRA KERNELS

Tania Stathaki and Anne Scohyers

Signal Processing and Digital Systems Section
Imperial College, UK
tania@ic.ac.uk

ABSTRACT

A novel approach is taken for the estimation of the parameters of a Volterra model, which is based on constrained optimisation. The equations required for the determination of the Volterra kernels are formed entirely from the second and higher order statistical properties of the "output" signal to be modelled and can therefore be classed as blind in nature. These equations are highly nonlinear and their solution is achieved through a judicious use of reliably measured statistical features of the signal to be modelled, in conjunction with appropriate constraints and penalty functions. Examples are given to illustrate the method and it is evident from those that this novel approach is producing useful results in contexts that have been hitherto unattainable.

1. INTRODUCTION

Considerable attention has been focused in published papers (for example see [1-5]), on the subject of nonlinear signal modelling by means of Volterra representations. The case in which the input is assumed to be known, i.e. the non blind form of the problem can be readily resolved by using the cross-cumulants between the input and the output [1-3].

However, in a realistic form of the problem, only the output is observable, and thus for any subsequent modelling we must use only the measured output data. If the input is a general random process it is extremely difficult if not impossible to find closed-form expressions for the Volterra kernels. So far, a fundamental assumption underlying many of the approaches to the problem involve the fact that the "input" is assumed to be a stationary random process with Gaussian statistics, an assumption which allows a substantial simplification of the relevant mathematics. It has been shown that, for a zero mean white Gaussian input, nonlinear expressions for the linear and quadratic transfer functions are given in terms of various spectral moments up to third order (i.e. the bispectrum) of the signal [4-5].

The outstanding difficulty in such modelling is related to the quest for finding solutions for these highly nonlinear equations that yield the Volterra kernels.

In this contribution we employ a quadratic Volterra filter form for the modelling of one dimensional signals.

In recent studies [4-5] we used the nonlinear equations that yield the Volterra parameters to form an unconstrained optimisation problem which was solved using Lagrange Programming Neural Networks (LPNN) [6-7].

In [8] another approach is proposed for the determination of the Volterra kernels which is now based on constrained optimisation using again LPNN. In [8] the new contribution is that we pay particular attention to the reliability of the statistical measures used in the process. Indeed, the second order moments are known to be more reliable than higher moments. Use of this fact is made in the construction of the constraints of the optimisation problem.

In this paper we pay again particular attention to the second order statistical measures but in a different sense. More specifically we use the equations that relate the unknown parameters of the model with the autocorrelations of the signal to form a penalty function [9]. This function is incorporated to the cost function and yields a so called Augmented Lagrangian function. The method presented in this study belongs to a class of optimisation methods called penalty-transformation methods. These seem to provide better results to the Volterra parameter estimation problem than the Lagrange methods, concerning convergence and computational properties.

2. PRELIMINARIES

We represent the signal as the output of a non linear time invariant causal system driven by noise $x[n]$. The Volterra representation of the input/output relationship is given by

$$y[n] = h_o + \sum_i h_1[i]x[n-i] + \sum_i \sum_j h_2[i,j]x[n-i]x[n-j]$$

$$+ \sum_i \sum_j \sum_k h_3[i,j,k]x[n-i]x[n-j]x[n-k] + \ldots$$

where $h_1[i]$, $h_2[i,j]$, $h_3[i,j,k],\ldots$ are the linear, quadratic, cubic etc. filter weights or kernels respectively, h_o is a constant term whose value depends on the input $x[n]$ and $0 \le i,j,k \le N-1$ where N denotes the filter length [4-5].

The above is the discrete version of the functional form of a nonlinear continuous time invariant system with memory which was first studied by V. Volterra in the 1880's as a generalisation of the Taylor series of a function.

$$y[t] = h_o + \int_{-\infty}^{+\infty} h_1[\tau_1]x[t-\tau_1]d\tau_1$$

$$+ \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} h_2[\tau_1,\tau_2]x[t-\tau_1]x[t-\tau_2]d\tau_1 d\tau_2$$

$$+\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} h_3[\tau_1,\tau_2,\tau_3]x[t-\tau_1]x[t-\tau_2]x[t-\tau_3]d\tau_1 d\tau_2 d\tau_3+\ldots$$

As a linear, time-invariant (LTI) system is completely characterised by its unit impulse response, so a nonlinear system which can be represented by a Volterra series is completely characterised by its Volterra kernels. Also, arguing in a manner similar to that for linear systems, it can be shown that the nonlinear system is causal if and only if $h_n[\tau_1,\tau_2,\ldots,\tau_n]=0$, for any $\tau_j<0,\ j=1,\ldots,n$.

We consider in this paper without loss of generality symmetric Volterra kernels only. A symmetric kernel is a symmetric function of its arguments so that for n arguments $\tau_1,\tau_2,\ldots,\tau_n$ there are $n!$ possible interchanges that leave $h_n[\tau_1,\tau_2,\ldots,\tau_n]=0$ unchanged. Specifically, for $n=2$, $h_2[\tau_1,\tau_2]$ is symmetric if $h_2[\tau_1,\tau_2]=h_2[\tau_2,\tau_1]$.

However, in spite of the generality of the functional series expansion of nonlinear systems, relatively few researchers have attempted to identify from outputs alone any practical systems based on this representation. This can be attributed to the formidable amount of computation required and the difficulties associated with the identification of the system's kernels. Indeed, identification of systems which contain anything higher than second-order kernels is a very difficult task because of this excessive amount of computation.

The second order Volterra filter is given by the following relationship:

$$y[n]=\sum_i a[i]x[n-i]\ +\sum_i\sum_j b[i,j]x[n-i]x[n-j]$$

where we assumed that $h_o=0$ without loss of generality [5]. By assuming that the input signal $x[n]$ is a discrete, stationary, zero mean, white Gaussian process, the output process is also discrete, stationary but non-Gaussian process and not necessarily zero mean. To simplify the expressions we assume that the output is zero mean in which case the following condition $\sum_i b[i,i]=0$ must hold.

3. SECOND ORDER STATISTICAL ANALYSIS

In this section the second order statistical analysis of the output will be considered. The autocorrelation function of the real process $y[n]$ is given by $R[k]=E\{y[n]y[n+k]\}$ and in view of the second order Volterra model can be written as:

$$R[k]=\sum_i a[i]E\{x[n-i]y[n+k]\}$$
$$+\sum_i\sum_j b[i,j]E\{x[n-i]x[n-j]y[n+k]\} \qquad (1)$$

The terms of equation (1) involve averaging over the product of one, two, three and four Gaussian random variables. It is known that the average of the product of an odd number of zero-mean jointly Gaussian random variables is identically zero irrespective of their mutual correlation. Moreover, the average of the product of an even number of zero-mean jointly Gaussian random variables is equal to the summation over all distinct ways of partitioning the random variables into products of averages of pairs. For example, if x_1,x_2,x_3,x_4 are zero-mean jointly Gaussian random variables, then:

$$E\{x_1x_2x_3\}=0 \qquad (2)$$

$$E\{x_1x_2x_3x_4\}=$$
$$E\{x_1x_2\}E\{x_3x_4\}+E\{x_1x_3\}E\{x_2x_4\}+E\{x_1x_4\}E\{x_2x_3\} \qquad (3)$$

With (2) and (3), $R[k]$ reduces to the form:

$$R[k]=\beta\sum_i a[i]a[i+k]\ +2\beta^2\sum_i\sum_j b[i,j]b[i+k,j+k] \qquad (4)$$

where β is the variance of the input driving noise.

The autocorrelation function given by (4) is not sufficient to solve the problem because the number of unknowns present in these equations is much greater than the number of useful samples of $R[k]$. For example for the one dimensional case if the kernels $a[i]$ and $b[i,j]$ have length N, then the number of equations provided is N, while the number of unknowns is $N(N+3)/2+1$. However, additional information can be provided by examining higher order statistics [5].

4. THIRD ORDER STATISTICAL ANALYSIS

In this section additional information is shown to be provided by examining the higher order statistics of the output, and this information can be employed towards the solution of the nonlinear equation.

If we define $M[k,l]$ to be the third order moment sequence of $y[n]$, then [5]:

$$M[k,l]==E\{y[n]y[n+k]y[n+l]\} \qquad (5)$$

In the following, the third-order moment sequence of the second order Volterra filter is derived. First we use the following symbols:

$$G_1[k]=\sum_i a[i]x[n-i+k] \qquad (6)$$

$$G_2[k]=\sum_i\sum_j b[i,j]x[n-i+k]x[n-j+k] \qquad (7)$$

Based on (6), (7) one can easily expand (5) in the following compact form:

$$M[k,l]=\ E\{G_1[0]G_1[k]G_1[l]\ +G_1[0]G_1[k]G_2[l]\ +G_1[0]G_2[k]G_1[l]$$
$$+G_1[0]G_2[k]G_2[l]\ +G_2[0]G_1[k]G_1[l]\ +G_2[0]G_1[k]G_2[l]$$
$$+G_2[0]G_2[k]G_1[l]\ +G_2[0]G_2[k]G_2[l]\} \qquad (8)$$

The first, fourth, sixth and seventh terms of (8) involve averaging over an odd number of zero-mean jointly Gaussian random variables. Therefore are identically zero. Equation (8) then becomes:

$$M[k,l]=\ E\{G_1[0]G_1[k]G_2[l]\ +G_1[0]G_2[k]G_1[l]\ +G_2[0]G_1[k]G_1[l]$$
$$+G_2[0]G_2[k]G_2[l]\} \qquad (9)$$

Each term of (9) involves averaging over an even number of zero-mean jointly Gaussian random variables. Keeping in mind the procedure we described in the previous paragraph, one can decompose the average of the product of an even number of jointly Gaussian random variables into a summation of products of averages of pairs. The first term of (9) (not using the fact that $b[i,j]$ is a symmetric kernel), can then be written as follows:

$$E\{G_1[0]G_1[k]G_2[l]\} = E\{(\sum_i a[i]x[n-i])\ (\sum_i a[i]x[n-i+k])\ (\sum_i\sum_j b[i,j]x[n-i+l]x[n-j+l])\}$$

$$= 2\beta^2 \sum_i\sum_j a[i+l]a[j+l-k]b[i,j]$$

Now we define $\phi_1[k,l]$ to be as follows:

$$\phi_1[k,l] = \sum_i\sum_j a[i]a[j]b[i+k,j+l]$$

It can be proven that

$$E\{G_1[0]G_1[k]G_2[l]\} = 2\beta^2\phi_1[l,l-k] \qquad (10)$$

Similarly, one can show that:

$$E\{G_1[0]G_2[k]G_1[l]\} = 2\beta^2\phi_1[k,k-l] \qquad (11)$$

$$E\{G_2[0]G_1[k]G_1[l]\} = 2\beta^2\phi_1[-k,-l] \qquad (12)$$

The fourth term of (9) is quite different from the first three terms. It involves averaging over the product of four Gaussian random variables as well as averaging over the product of six Gaussian random variables. The latter can be broken into the sum of fifteen terms, where each term involves a product of three averages of distinct pairs of random variables. An analysis of the procedure is given in the Appendix.

By doing so and defining:

$$\phi_2[x,y,z] = \sum_i\sum_j\sum_k b[i,j]b[i+x,k+y]b[j+z,k]$$

we obtain:

$$E\{G_2[0]G_2[k]G_2[l]\} = 8\beta^3\phi_2[l,l-k,k] \qquad (13)$$

which is valid only when $\sum_i b[i,i] = 0$ or $b[i,i] = 0,\ \forall i$. In other case equation (13) is more complicated.

We replace (10), (11), (12), (13) in (9) and we obtain $M[k,l]$. It is now possible to use (4) in conjunction with (9) to provide a sufficient number of nonlinear equations required to solve for the unknown system parameters a and b and the variance of the driving noise β.

5. LAGRANGE PROGRAMMING NEURAL NETWORKS (LPNN)

LPNN is a neural network primarily designed for general nonlinear programming. The methodology is based on the well-known Lagrange multiplier method for general constrained optimisation problems. The essential components of the approach are as follows [6-7].

Consider the following constrained nonlinear programming problem:

Minimise $X(f)$

Subject to $Y(f) = 0$

where $X: R^n \to R$ and $Y: R^n \to R^m$ are given functions and $m \le n$. The components of Y are denoted $Y_1,\dots,Y_m$. We can define the Lagrangian function $L: R^{n+m} \to R$ by

$$L(f,\lambda) = X(f) + \lambda^t Y(f)$$

where $\lambda \in R^m$ is referred to as the Lagrangian multiplier.

The dynamic equations of the LPNN are then defined as:

$$\frac{df}{dt} = -\nabla_f L(f,\lambda) \qquad (14)$$

$$\frac{d\lambda}{dt} = \nabla_\lambda L(f,\lambda) \qquad (15)$$

where

$$\nabla_f L(f,\lambda) = \left[\frac{dL}{df_1},\frac{dL}{df_2},\dots,\frac{dL}{df_n}\right]^t \text{ and}$$

$$\nabla_\lambda L(f,\lambda) = \left[\frac{dL}{d\lambda_1},\frac{dL}{d\lambda_2},\dots,\frac{dL}{d\lambda_m}\right]^t, \text{ respectively.}$$

If the network is physically stable, then the equilibrium point (f^*,λ^*) defined by $\frac{df}{dt} = 0$ and $\frac{d\lambda}{dt} = 0$, satisfies the equations:

$$\nabla_f L(f^*,\lambda^*) = \nabla X(f^*) + \nabla Y(f^*)^t\lambda^* = 0$$

$$\nabla_\lambda L(f^*,\lambda^*) = Y(f^*) = 0$$

and thus provides a Lagrange solution to the optimisation problem.

Equations (4) furnish a set of relationships for the autocorrelations while (9) provide the third order moments and both of those quantities can be estimated by standard means from a given signal. We are seeking for the parameters $\{a[i]\}$, $\{b[i,j]\}$ and β of a Volterra model that would produce these second and third order moments.

The problem is therefore as follows: Given the autocorrelation estimates $\rho[k]$ and the third order moments estimates $\mu[k,l]$ obtained from the signal directly, to determine $\{a[i]\}$, $\{b[i,j]\}$ and β.

5.1 Unconstrained Optimisation

In [4-5] we present a solution to the above problem in the LPNN sense, which starts initially with a suitable cost function such as the one given below

$$L(f) = \sum_i(\rho[i] - R[i,f])^2 + \sum_i\sum_j(\mu[i,j] - M[i,j,f])^2$$

From this cost function and in accordance with the above formulations, the LPNN dynamic equations may be set up as in (14). Notice that no constraints have been incorporated since the above formulation allows any values of $\{a[i]\}$ and $\{b[i,j]\}$ to exist. Thus the Lagrange parameters are set to zero, or the corresponding Lagrange neurons are clamped to zero level.

In the above equation $f = (\underline{a},\underline{b},\beta)$ is a vector formed by the unknown parameters of the Volterra model and the unknown variance of the driving noise. The signal flow graph of these equations describes the required dynamic neural network structure the steady state of which delivers the solution.

5.2 Constrained Optimisation

In [8] the minimisation is carried out under certain constraints which we have chosen in a way to reflect the accuracy of our measurements and of the estimation procedures. For finite duration signals autocorrelations are more accurately estimated than higher order moments, so a constrained nonlinear programming problem is formulated as follows.

Minimise:

$$L(f)=\sum_i\sum_j\left(\mu[i,j]-M[i,j,f]\right)^2$$

subject to

$$\rho[i]=R[i,f]$$

In this form we have a constrained optimisation problem for which we form the following Lagrange function

$$L(f,\lambda)=L(f)+\sum_i\lambda_i\left(\rho[i]-R[i,f]\right)$$

The dynamic equations of the LPNN (which are the update equations for f and λ) are now defined as in (14) and (15).

The stability of the neural network and the optimality of the solution are guaranteed under some regularity and convexity conditions [6],[7]. The method presented in [8] provides a solution closer to the optimum than the unconstrained method in [4-5].

5.3 Penalty Methods

In this study we pay again particular attention to the second order statistical measures but in a different sense. More specifically we use the equations that relate the unknown parameters of the model with the autocorrelations of the signal to form a penalty function [7]. This function is incorporated to the cost function and yields a so called Augmented Lagrangian function. The problem now has as follows:

$$L(f,\lambda)=L(f)+\sum_i\lambda_i\left(\rho[i]-R[i,f]\right)+c\sum_i\left(\rho[i]-R[i,f]\right)^2$$

where $\{c_k\}$ is a penalty parameter sequence satisfying

$$0<c_k<c_{k+1}\ \forall k,\ c_k\rightarrow\infty$$

The development of the above method was motivated by the concept of maintaining implicit control over the violations of constraints by penalising the objective function at points which violate or perhaps tend to violate the constraints.

6. SIMULATIONS

In the simulations presented below we apply the new approach and compare the results with the constrained approach in [8]. We use two different versions of the same synthetic one dimensional signal which for our purposes is described by a quadratic of the following form:

$$y[n]=x[n]+1.8x[n-1]+0.5x[n]x[n-1]$$

The first version is of size 1000 samples and the second of size 500 samples. We apply both the constrained optimisation approach introduced in [8] and the constrained optimisation with penalty term approach presented in this paper for the same signal. We are dealing with nonlinear and nonconvex functions so we might encounter difficulties concerning the convergence of the algorithm. In this work we repeat the same experiments starting from different initial points for the unknown parameters, chosen to be in the attraction basin of the global solution that can be approximately determined using simulated annealing algorithms. The first table below shows the solutions arising from the method which was presented in [8] for 1000 samples. The second table contains the results for the same experiments but using the procedure described in this study. The third and fourth tables correspond to the same results but for 500 samples.

It is observed from the tables below that the new approach yields improved results for both signals as expected also theoretically compared to the method in [8] and to the extent in [4-5]. These results arise from a large number of tests involving different signals.

parameter	real value	mean estimated value
$a[0]$	1	1.12
$a[1]$	1.8	1.83
$b[0,1]$	0.5	0.5075
β	1.5532	1.504

parameter	real value	mean estimated value
$a[0]$	1	1.0265
$a[1]$	1.8	1.825
$b[0,1]$	0.5	0.506
β	1.5532	1.5165

parameter	real value	mean estimated value
$a[0]$	1	1.0949
$a[1]$	1.8	1.8379
$b[0,1]$	0.5	0.5839
β	1.5532	1.57

parameter	real value	mean estimated value
$a[0]$	1	1.037
$a[1]$	1.8	1.831
$b[0,1]$	0.5	0.55
β	1.5532	1.562

REFERENCES

[1] T. Koh and E. J. Powers, "Second-order Volterra filtering and its application to nonlinear system identification", *IEEE, ASSP*, Vol. 33, No. 6. Dec. 1985

[2] K. I. Kim and E. J. Powers, "A digital method of modeling quadratically nonlinear systems with a general random input", *IEEE Trans. on ASSP*, Vol. 36, No. 11, November 1988.

[3] S. P. Parker and F. A. Perry, "A discrete ARMA model for nonlinear system identification", *IEEE Trans. on Circuits and Systems*, Vol. CAS-28, No. 3, March 1981.

[4] Tania Stathaki, "Two dimensional blind Volterra signal modelling", *Proc. of the IEEE Workshop on Higher Order Statistics*, 12-14 June 1995, SPAIN.

[5] Tania Stathaki, "Blind Volterra signal modelling", *Trans. of the ICASSP 96*, vol. 3, pages 1601-1604, May 7-10, 1996, Atlanta, Georgia, USA.

[6] S. Zhang and A. G. Constantinides "Lagrange programming neural networks", *IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing*, Vol. 39, No. 7., July 1992.

[7] D. P. Bertsekas, *Constrained Optimisation and Lagrange Multipliers Method*, Academic Press, New York, 1982.

[8] Tania Stathaki, "Blind Volterra modelling using constrained optimisation", *Proc. of the Asilomar Conference on Signals, Systems and Computers*, November 3-6 1996.