PERGAMON

Neural Networks 13 (2000) 255–266

Neural Networks

www.elsevier.com/locate/neunet

Contributed article

A novel network for nonlinear modeling of neural systems with arbitrary point-process inputs

Konstantinos Alataris, Theodore W. Berger*, Vasilis Z. Marmarelis

Department of Biomedical Engineering, University of Southern California, OHE 500, Los Angeles, CA 90089-1451, USA

Received 20 March 1998; received in revised form 6 October 1999; accepted 6 October 1999

Abstract

This paper address the issue of nonlinear model estimation for neural systems with arbitrary point-process inputs using a novel network that is composed of a pre-processing stage of a Laguerre filter bank followed by a single hidden layer with polynomial activation functions. The nonlinear modeling problem for neural systems has been attempted thus far only with Poisson point-process inputs and using cross-correlation methods to estimate low-order nonlinearities. The specific contribution of this paper is the use of the described novel network to achieve practical estimation of the requisite nonlinear model in the case of arbitrary (i.e. non-Poisson) point-process inputs and high-order nonlinearities. The success of this approach has critical implications for the study of neuronal ensembles, for which nonlinear modeling has been hindered by the requirement of Poisson process inputs and by the presence of high-order nonlinearities. The proposed methodology yields accurate models even for short input–output data records and in the presence of considerable noise. The efficacy of this approach is demonstrated with computer-simulated examples having continuous output and point-process output, and with real data from the dentate gyrus of the hippocampus. © 2000 Published by Elsevier Science Ltd. All rights reserved.

Keywords: Nonlinear modeling; Point-process inputs; Neural networks; Laguerre functions; Volterra model; Neural systems

1. Introduction

The task of nonlinear modeling of physiological systems is very challenging because of the immense variety of nonlinearities in physiological systems and the diverse requirements of different applications. This task becomes more demanding when we lack information about the internal workings of the system and we seek to obtain "empirical" mathematical models on the basis of experimental input–output data (Marmarelis & Marmarelis, 1978; Marmarelis, 1987, 1989, 1994).

In this context of identification of nonlinear physiological systems, the theory of functional expansions (Volterra or Wiener series) (Schetzen, 1980) and Gaussian white noise (GWN) test inputs have been widely used (Barrett, 1963; Marmarelis & Marmarelis, 1978; Volterra, 1930). In this context, the most commonly used method to date is based on cross-correlation and requires input whiteness—a condition met by the Poisson process for point-process inputs (Berger, Robinson, Port & Sclabassi, 1987; Krausz, 1975). However in many actual applications of the Volterra–Wiener approach, the prevailing conditions are less than ideal. Critical among them are: (a) deviations from input whiteness that may be caused by experimental necessity or inadvertent stimulus distortion; (b) extraneous noise that contaminates the experimental data; (c) occasional presence of non-stationarities that make it imperative to obtain accurate results from short experimental records; and (d) long system memory requirements that create a heavy computational burden when the conventional cross-correlation method is used. The traditional cross-correlation technique (CCT) (Lee & Schetzen, 1965) and its many variants over the last 30 years require whiteness of the system input and yield a set of kernels that correspond to the orthogonal functional expansion associated with the specific input, which are distinct from the Volterra kernels and depend on the input characteristics, e.g. for GWN inputs we obtain the Wiener kernels.

Therefore new techniques are needed that address these restrictive requirements. For instance, the Laguerre expansion technique (LET) has been recently introduced to obviate the need for input whiteness and reduce the computational burden for long memory systems, although it may still impose certain restrictions (e.g. sufficiently broadband inputs and complete estimated models) (Marmarelis, 1993). In addition, LET is robust in the presence of noise and can yield accurate estimates from short experimental data

* Corresponding author. Tel.: +1-213-740-8017; fax: +1-213-740-6174.
E-mail address: berger@bmsrs.usc.edu (T.W. Berger).

PII: S0893-6080(99)00092-1

records, avoiding the pitfalls of non-stationarity in the experimental preparation. In spite of its many advantages, the application of LET is still limited to low-order systems (up to third) because of the computational burden associated with the dimensionality of high-order kernels. This has prompted the introduction of the principal dynamic mode methodology that seeks to make practical the estimation of high-order nonlinearities (Marmarelis & Orme, 1993).

If the input signal is non-white, kernel estimation must be based on minimization of the output-prediction mean-square error (MSE), whereby the kernel estimation task is converted into a parameter estimation problem via least-squares fitting. Since the unknown discrete kernel values enter linearly in the input–output equation (i.e. Volterra model), their estimation is possible through linear regression in one of its many implementations (e.g. ordinary or generalized least-squares, instrumental variables, singular-value decomposition, QR decomposition, etc.) most suitable for a given application. The use of Artificial Neural Nets (ANN) and adaptive estimation techniques (used for their training) also have been proposed for the same purpose (Chen, Billings & Grant, 1990; Marmarelis & Zhao, 1997; Wray & Green, 1994).

The proposed methodology is a hybrid of Laguerre expansions and ANN adaptive estimation, as detailed below. It is the performance of this network (termed herein the Laguerre–Volterra network "LVN") with respect to nonlinear modeling of neural systems that is the subject of this paper. The main objective is to show that the LVN can practically estimate high-order nonlinear models of neural systems with arbitrary (i.e. non-Poisson) point-process inputs. In terms of modeling performance, the proposed LVN combines the strengths of the traditional methods: compactness of representation from the Laguerre expansion, affinity with Volterra modeling for biological interpretation, and ease of estimation of high-order models with arbitrary inputs using the efficient adaptive estimation algorithms for training multi-layer perceptrons.

2. Methodology

For a causal system, the relation between the known input-output signals can be seen as a mapping of the past and present values of the input signal onto the present value of the output signal. The use of a mathematical functional can be used to represent this mapping:

$$y(t) = F[x(\tau),\ \tau \leq t] \tag{1}$$

where x is the input signal, y the output signal and F the functional representing the system. When the system is time-invariant (stationary), the functional F remains unchanged through time. The modeling goal is to find an explicit mathematical description of the system functional F.

In the Volterra approach (Volterra, 1930), a functional expansion of the system functional F is used for this purpose. For discretized input–output data, the general model for causal, stable, nonlinear, time-invariant systems is given by the Volterra series expansion:

$$\begin{aligned} y(n) = k_0 &+ \sum_{m=0}^{M} k_1(m)x(n-m) \\ &+ \sum_{m_1=0}^{M}\sum_{m_2=0}^{M} k_2(m_1,m_2)x(n-m_1)x(n-m_2) \\ &+ \sum_{m_1=0}^{M}\sum_{m_2}^{M}\sum_{m_3=0}^{M} k_3(m_1,m_2,m_3)x(n-m_1)x(n-m_2) \\ &\times x(n-m_3) + \dots \end{aligned} \tag{2}$$

where n is the discrete time, $x(n)$ the input data sequence, $y(n)$ the output data sequence and $\{k_i\}$ are the Volterra kernels to be estimated from the known input–output data. For systems with finite memory M, the input–output relation can be viewed as a nonlinear mapping of the input epoch $\{x(n), x(n-1), \dots, x(n-M)\}$ on to the output value $y(n)$, for every value of discrete time n. Thus, the Volterra model is constituted by a hierarchy of functional terms mapping an epoch $[n,\ n-M]$ of the input signal onto the output present value $y(n)$.

The Volterra kernels $\{k_i\}$ describe the nonlinear dynamics of the system (i.e. fully characterize the nonlinear input–output mapping) at each order of nonlinearity and constitute a complete and canonical representation of any stable system whose output changes infinitesimally in response to an infinitesimal change of the input signal (Rugh, 1981). For a uniformly bounded input, the output remains uniformly bounded if and only if the system kernels are absolute-summable and form a convergent series (Volterra class of systems). The kernels are symmetric (i.e. invariant to any permutation of their arguments) and for causal systems are zero for negative values of their arguments (Volterra, 1930; Wiener, 1958).

Estimation of the Volterra kernels $\{k_i\}$ can be achieved by various methods, including the use of ANN training (Wray and Green, 1994), and the equivalence between Volterra models and feedforward ANN has been shown (Marmarelis and Zhao, 1997). Since the ANN is trained with the available input-output data, the training task does not place any specific requirements on the input, although an ergodic and spectrally rich input is expected to yield better training results (i.e. capable of generalization).

As indicated above, the proposed methodology employs an ANN with a single hidden layer and polynomial activation functions, receiving as input the outputs of a Laguerre filter bank that pre-processes the stimulus data. This model, termed the Laguerre–Volterra network (LVN), is a valid representation of Volterra systems, as originally suggested by Wiener (1958) and elaborated by Watanabe and Stark

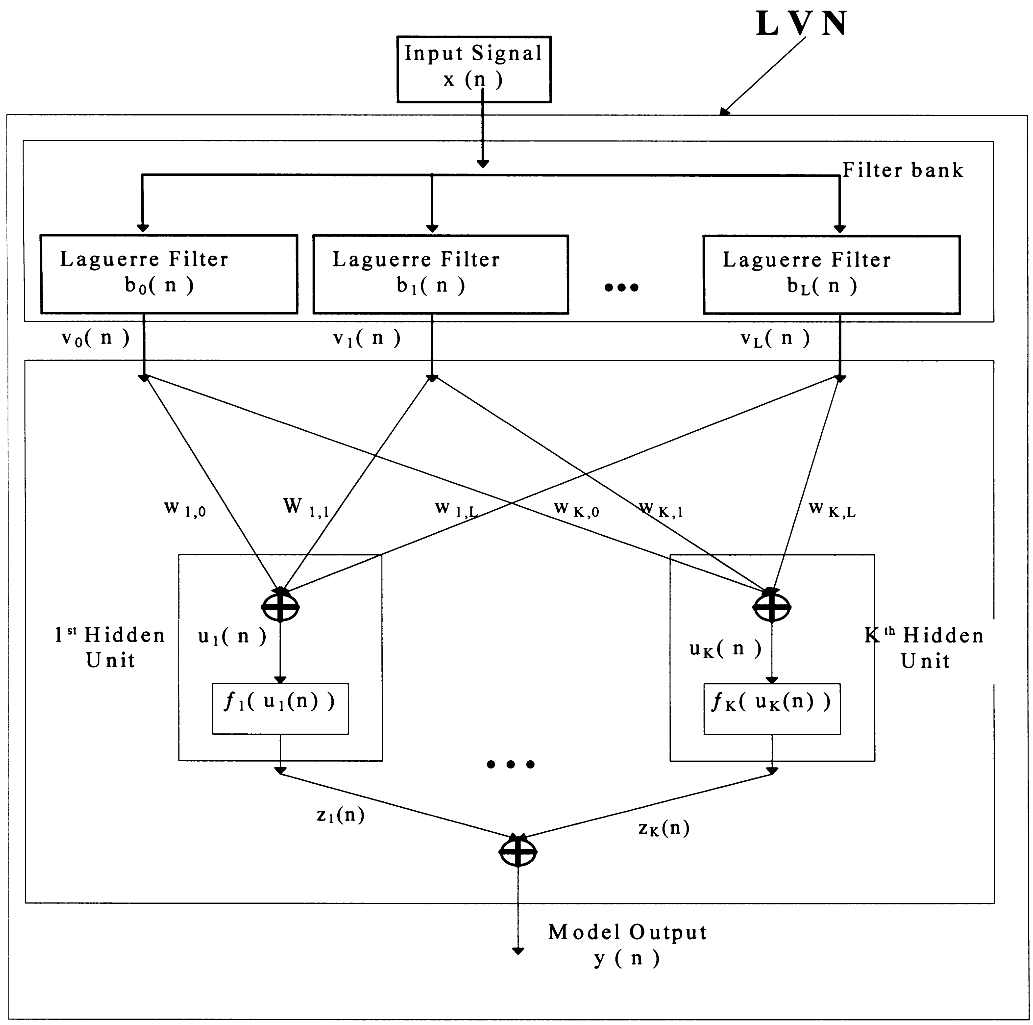

Fig. 1. The proposed model (LVN) employs an ANN with a single hidden layer and polynomial activation functions, receiving as input the outputs of a Laguerre filter bank that pre-processes the input data.

(1975) and Schetzen (1980). This is evident when we consider kernel expansions on the complete orthonormal Laguerre basis $\{b_j(m)\}$ as (Marmarelis, 1993):

$$k_r(m_1,\ldots,m_r) = \sum_{j_1,\ldots,j_r} a_r(j_1,\ldots,j_r) b_{j_1}(m_1),\ldots,b_{j_r}(m_r) \tag{3}$$

and by substitution in the Volterra series of Eq. (2), we obtain:

$$y(n) = k_0 + \sum_{j_1} a_1(j_1) v_{j_1}(n) + \sum_{j_1,j_2} a_2(j_1,j_2) v_{j_1}(n) v_{j_2}(n) + \cdots \tag{4}$$

where,

$$v_j(n) = \sum_m b_j(m) x(n-m) \tag{5}$$

Expression (4) is a multinomial expansion of the output signal on the variables $\{v_j(n)\}$, which can be viewed as the outputs of Laguerre filters receiving input $x(n)$. Eqs. (4) and (5) describe the equivalent block-structured Volterra model composed of the Laguerre filter bank cascaded with a zero-memory (static) nonlinearity. This model can be called the "reduced Wiener model" to honor its original proponent. The goal of this study is to demonstrate that the LVN can be used in a practical context to yield models of the general class of Volterra systems with arbitrary point-process inputs.

Fig. 1 shows the LVN architecture: the Laguerre filters $\{b_j\}$ receive the stimulus data $x(n)$ and generate (by convolution) the intermediate variables $\{v_j\}$, $j = 0, 1, \ldots, L$, that constitute the input vector (at each time n) for the hidden layer comprised of K hidden units $\{H_i\}$, $i = 1, \ldots, K$. The outputs $\{z_i(n)\}$ of the hidden units are summed to generate the output $y(n)$ of the LVN. The basic mathematical relations for the LVN are:

$$v_j(n) = \sum_{m=0}^{M} b_j(m) x(n-m) \qquad (j = 0, 1, \ldots, L) \tag{6}$$

$$u_i(n) = \sum_{j=0}^{L} w_{j,i} v_j(n) \qquad (i = 1, \ldots K) \tag{7}$$

$$z_i(n) = c_{i,0} + c_{i,1} u_i(n) + \cdots + c_{i,Q}[u_i(n)]^Q \tag{8}$$

$$y(n) = \sum_{i=1}^{K} z_i(n) \tag{9}$$

Supervised training of the LVN is performed on the $K(L+1)$ weights $\{w_{j,i}\}$ and on the $K(Q+1)$ coefficients $\{c_{i,q}\}$ of the K distinct polynomial activation functions of Qth degree. This training can be performed by means of gradient descent methods (commonly known as delta rule or error back-propagation) of the quadratic norm of the output MSE:

$$E(n) = \tfrac{1}{2}[\tilde{y}(n) - y(n)]^2 \tag{10}$$

where $\{\tilde{y}(n)\}$ are the actual output observations (data).The specifics of such commonly used training methods can be found in various books on artificial neural networks (Haykin, 1994; Zurada, 1992).

One novelty of the LVN architecture is the polynomial activation functions, the use and training of which recently has been reported in connection with Volterra models (Marmarelis & Zhao 1997). This polynomial form of the activation functions allows an easy transition between ANN and the Volterra models, as elaborated below. It is evident that the LVN can be used as an equivalent model for the Volterra class of systems of finite order Q.

Although the filters $\{b_j(m)\}$ in the filter bank of the LVN can be any complete basis, we chose the discrete-time orthonormal Laguerre basis based on previous studies that have demonstrated its desirable properties for the expansion of Volterra kernels for many physiological systems (Marmarelis 1993; Watanabe & Stark, 1975). Laguerre expansions and filters have been used for the identification of linear dynamic systems (Mäkilä, 1990a,b; Wahlberg, 1991; Wahlberg & Mäkilä, 1995).

The impulse response functions of the discrete-time Laguerre filters are given by (Broome, 1965; Ogura, 1985):

$$b_j(m) = \alpha^{(m-j)/2}(1-\alpha)^{1/2} \sum_{k=0}^{j} (-1)^k \binom{m}{k}\binom{j}{k} \alpha^{j-k}(1-\alpha)^k$$

$$(0 \le m \le M) \tag{11}$$

where α is the discrete-time Laguerre parameter $(0\angle\alpha\angle 1)$ that regulates the exponential decay of the Laguerre basis suitable for the relaxation characteristics of a given system (Marmarelis, 1993). The proper selection of α is critical for successful application of this method. Along with the determination of the key parameters M, L, K, Q, it is based on trial and error, i.e. we increase the parameter values until the resulting reduction in the MSE of the output prediction is below a predetermined threshold.

Note that, instead of evaluating the convolution in Eq. (6) to compute the variables $\{v_j(n)\}$, we can use more efficiently the recursive formula (Marmarelis, 1993):

$$v_j(n) = \sqrt{\alpha} v_j(n-1) + \sqrt{\alpha} v_{j-1}(n) - v_{j-1}(n-1) \tag{12}$$

initialized by:

$$v_0(n) = \sqrt{\alpha} v_0(n-1) + \sqrt{1-\alpha}\, x(n) \tag{13}$$

for $n = 1, \ldots, N$, where N is the total number of input datapoints. In addition to the computational efficacy of this recursive method, an additional advantage is that no assumptions are made about the system memory M and the only memory constraint is the data record length.

Using the basic Eqs. (6)–(9) of the LVN, we can express the output in terms of the input as the Volterra model:

$$\begin{aligned} y(n) = {} & \sum_{i=1}^{K} c_{i,0} + \sum_{i=1}^{K}\sum_{j=0}^{L}\sum_{m=0}^{M} c_{i,1} w_{j,i} b_j(m) x(n-m) \\ & + \sum_{i=1}^{K}\sum_{j_1=0}^{L}\sum_{j_2=0}^{L}\sum_{m_1=0}^{M}\sum_{m_2=0}^{M} c_{i,2} w_{j_1,i} w_{j_2,i} bj_1(m_1) bj_2(m_2) \\ & \times\ x(n-m_1)x(n-m_2) \\ & + \cdots \sum_{i=1}^{K}\sum_{j_1=0}^{L}\cdots\sum_{j_Q=0}^{L}\sum_{m_1=0}^{M}\cdots\sum_{m_Q=0}^{M} c_{i,Q} w_{j_1,i}\cdots w_{j_Q,i} b_{j_1}(m_1) \\ & \times \cdots b_{j_Q}(m_Q) x(n-m_1)\cdots x(n-m_Q) \end{aligned} \tag{14}$$

By comparing Eq. (14) with Eq. (2), we can derive the expressions for the system kernels in terms of the LVN parameters $\{w_{j,i}\}$ and $\{c_{i,q}\}$:

$$k_0 = \sum_{i=1}^{K} c_{i,0} \tag{15}$$

$$k_1(m) = \sum_{i=1}^{K}\sum_{j=0}^{L} c_{i,1} w_{j,i} b_j(m) \tag{16}$$

$$k_2(m_1, m_2) = \sum_{i=1}^{K}\sum_{j_1=0}^{L}\sum_{j_2=0}^{L} c_{i,2} w_{j_1,i} b_{j_1}(m_1) b_{j_2}(m_2) \tag{17}$$

$$\vdots$$

$$k_Q(m_1, \ldots, m_Q) = \sum_{i=1}^{K}\sum_{j_1=0}^{L}\cdots\sum_{j_Q=0}^{L} c_{i,Q} w_{j_1,i}\cdots w_{j_Q,i} b_{j_1}(m_1)\cdots b_{j_Q}(m_Q). \tag{18}$$

These expressions can be used to estimate Volterra kernels from trained LVN and resemble those derived by Wray and Green (1994) in their work linking the Volterra series with artificial neural networks. The network parameters that we adjust during training are the weights $\{w_{j,i}\}$ and the polynomial coefficients $\{c_{i,q}\}$ of the activation functions. The parameter updates take place at each presentation of a data pair $\{x(n), \tilde{y}(n)\}$ and are repeated sequentially through the data record. For the rth iteration at the nth data-point we have the parameter updates:

$$\Delta c_{i,q}(r) = \mu \Delta c_{i,q}(r-1) + \rho[\tilde{y}(n) - y(n)]u_i^q(n) \tag{19}$$

$$\Delta w_{j,i}(r) = \mu \Delta w_{j,i}(r-1) + \rho[\tilde{y}(n) - y(n)]v_j(n) \sum_{q=1}^{Q} q c_{i,q} u_i^{q-1}(n) \tag{20}$$

where μ is the momentum parameter, ρ is the learning constant, and $y(n)$ and $u_i(n)$ represent the most recently updated values. These updating expressions are derived following the conventional error back-propagation method for the LVN architecture.

In the case of point-process outputs, we can append a trainable threshold at the LVN continuous output. Then the post-threshold output is given by:

$$y_t(n) = \frac{\sigma}{1 + \mathrm{e}^{-\lambda(y(n)-\theta)}} \tag{21}$$

where σ is the saturation level of the sigmoidal threshold, λ is the slope of the transition between 0 and σ, and θ is the threshold (offset) level. All three parameters $(\sigma, \lambda, \theta)$ of the sigmoidal threshold can be adjusted during the training procedure to produce the best mean-square approximation of the observed point-process output. Following the error back-propagation rule, we need to calculate the gradient of the output error with respect to each threshold parameter. Also, the error gradient with respect to the LVN weights $\{w_{j,i}\}$ and the polynomial coefficients $\{c_{i,q}\}$ of the activation functions must be now multiplied with an additional factor:

$$\frac{\partial y_t(n)}{\partial y(n)} = \frac{\sigma\lambda\, \mathrm{e}^{-\lambda(y(n)-\theta)}}{[1 + \mathrm{e}^{-\lambda(y(n)-\theta)}]^2}. \tag{22}$$

The resulting updating expressions for the threshold parameters at the rth iteration are:

$$\Delta\sigma(r) = \mu\Delta\sigma(r-1) + \rho[\tilde{y}(n) - y_t(n)]\frac{1}{1 + \mathrm{e}^{-\lambda(y(n)-\theta)}} \tag{23}$$

$$\Delta\lambda(r) = \mu\Delta\lambda(r-1) + \rho[\tilde{y}(n) - y_t(n)]\frac{\sigma(y(n)-\theta)\,\mathrm{e}^{-\lambda(y(n)-\theta)}}{[1 + \mathrm{e}^{-\lambda(y(n)-\theta)}]^2} \tag{24}$$

$$\Delta\theta(r) = \mu\Delta\theta(r-1) + \rho[\tilde{y}(n) - y_t(n)]\frac{\sigma(-\lambda)\,\mathrm{e}^{-\lambda(y(n)-\theta)}}{[1 + \mathrm{e}^{-\lambda(y(n)-\theta)}]^2}. \tag{25}$$

3. Results

The efficacy of the LVN in modeling nonlinear neural systems is demonstrated here with computer simulated examples that include Poisson and non-Poisson point-process inputs. The cases of continuous output (graded potential) and point-process output (action potentials) are examined separately, because of their distinct signal processing characteristics and neurophysiological importance. The effect of output additive noise on estimation accuracy is examined in all those cases.

Many examples with systems/models of higher order and different forms of kernels have been successfully tested, and two illustrative examples are provided below for the continuous output case.

3.1. Continuous output

A relatively simple nonlinear system is selected for an initial computer simulated example. It is composed of the cascade of a linear filter and a static nonlinearity. The impulse response function of the linear filter is represented by a linear combination of three Laguerre functions for $\alpha = 0.7$ and coefficients: -0.90, $+0.33$, 0.70 for first, second and third order, respectively. The static nonlinearity is a third-degree polynomial with coefficients: 1.8, 3.5, -1.9 for the first, second and third degree terms, respectively.

Since the static nonlinearity is chosen to be a third-degree polynomial in the initial simulated example, the system output can be expressed as:

$$y(n) = \gamma_0 + \gamma_1[h(n) \otimes x(n)] + \gamma_2[h(n) \otimes x(n)]^2 + \gamma_3[h(n) \otimes x(n)]^3 \tag{26}$$

where $\otimes$ denotes convolution, $h(n)$ is the impulse response of the linear filter, and $\{\gamma_i\}$ are the polynomial coefficients of the static nonlinearity.

From Eq. (26) we can derive the Volterra kernels for this system:

$$k_0 = \gamma_0 \tag{27}$$

$$k_1(m) = \gamma_1 h(m) \tag{28}$$

$$k_2(m_1, m_2) = \gamma_2 h(m_1)h(m_2) \tag{29}$$

$$k_3(m_1, m_2, m_3) = \gamma_3 h(m_1)h(m_2)h(m_3). \tag{30}$$

As a measure of performance, we can evaluate the estimates of these kernels obtained by the LVN after training with the simulated input–output data. Another measure of performance can be the output prediction error, but here we chose the Volterra kernels to assess performance.

At first, the chosen system is simulated with a Poisson input of 512 datapoints containing 8 spikes. Since four Laguerre functions (including the zero-order one that is not present in the simulated system) suffice to represent

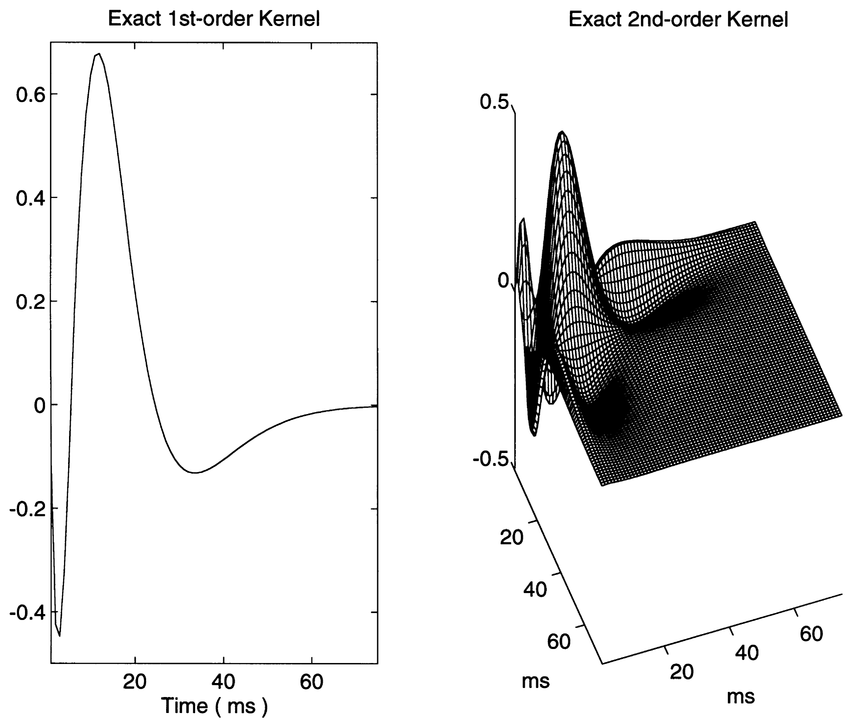

Fig. 2. The simulated nonlinear system is composed of the cascade of a linear filter and a static nonlinearity. The first and second order kernels of this system are shown over 75 lags.

this system, we select $L = 3$, one hidden unit ($K = 1$) and third-degree polynomial activation functions ($Q = 3$), which are the correct parameters for this system. The back-propagation algorithm utilizes a learning constant $\rho = 0.01$ and a momentum parameter $\mu = 0.1$. In the noise-free case, the training algorithm converges within 500 iterations to the precise parameter values. The 1st order and 2nd order kernel estimates are identical to the exact ones shown in Fig. 2.

This system is simulated next with an arbitrary (non-Poisson) point-process input having the same number of spikes and datapoints as before. The model parameters remain the same and the results are identical to those of the previous case (i.e. exact). This noise-free test was repeated for several arbitrarily chosen non-Poisson inputs and the algorithm always converged to the precise model parameter values.

The effect of noise was examined next by adding independent GWN to the output of the system when the input is an arbitrary (non-Poisson) point-process. The resulting kernels for a signal-to-noise ratio (SNR) of 0 dB are shown in Fig. 3, and demonstrate the robustness of this approach in the presence of output-additive noise. As another illustration, Fig. 4 shows the noisy output for a testing data-record (i.e. different from the one used for training) together with the model prediction and the noise-free output to allow comparison and demonstrate the exceptional robustness of this approach. Note that the noisy output of the testing data-record is contaminated by an independent segment of the stationary GWN process that also contaminates the noisy output of the training data-record.

In order to address possible concerns that the initial example is too simple and of low-order, the second simulated system is comprised of a linear filter with impulse response function:

$$h(n) = 0.9\,\mathrm{e}^{\frac{-n}{3}} + 1.5\,\mathrm{e}^{\frac{-n}{4}} - 2.7\,\mathrm{e}^{\frac{-n}{9}} + 0.9\,\mathrm{e}^{\frac{-n}{20}} \quad (31)$$

which is not constructed as linear combination of Laguerre functions, followed by a fifth-degree static nonlinearity with coefficients: $\gamma_0 = 0$, $\gamma_1 = 2$, $\gamma_2 = 5$, $\gamma_3 = 7$, $\gamma_4 = -4$, $\gamma_5 = 6$. The exact 1st and 2nd order kernels of this system are shown in Fig. 5. When an arbitrary (non-Poisson) point-process input is used as before, the resulting kernel estimates using 9 Laguerre functions with $\alpha = 0.7$ are identical to their exact counterparts and the LVN model prediction is identical to the system output, in the noise-free case. When GWN is added to the output for SNR = 0 dB, the obtained kernel estimates of 1st and 2nd order are shown in Fig. 6. Comparison with their exact counterparts shown in Fig. 5 corroborates the efficacy of the LVN modeling approach.

These results demonstrate the basic thesis of this paper, namely, that the proposed method is applicable for arbitrary point-process inputs and does not require Poisson inputs to yield accurate nonlinear models of neural systems. It must be noted that this is true so long as the point-process input does not have a very specific deterministic structure (e.g. a regular sequence of spikes with constant interspike interval would not be an effective input) and the LVN model parameters are not underspecified (e.g. less hidden units or

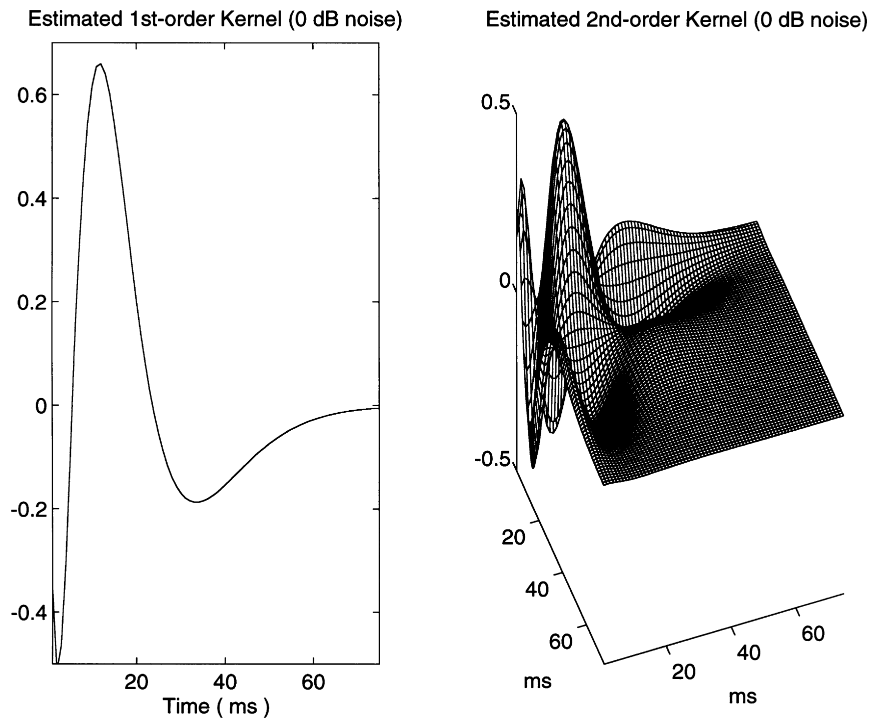

Fig. 3. The obtained 1st order and 2nd order kernel estimates for an arbitrary (non-Poisson) point-process input and independent Gaussian white noise added to the output resulting in the signal-to-noise ratio (SNR) of 0 dB.

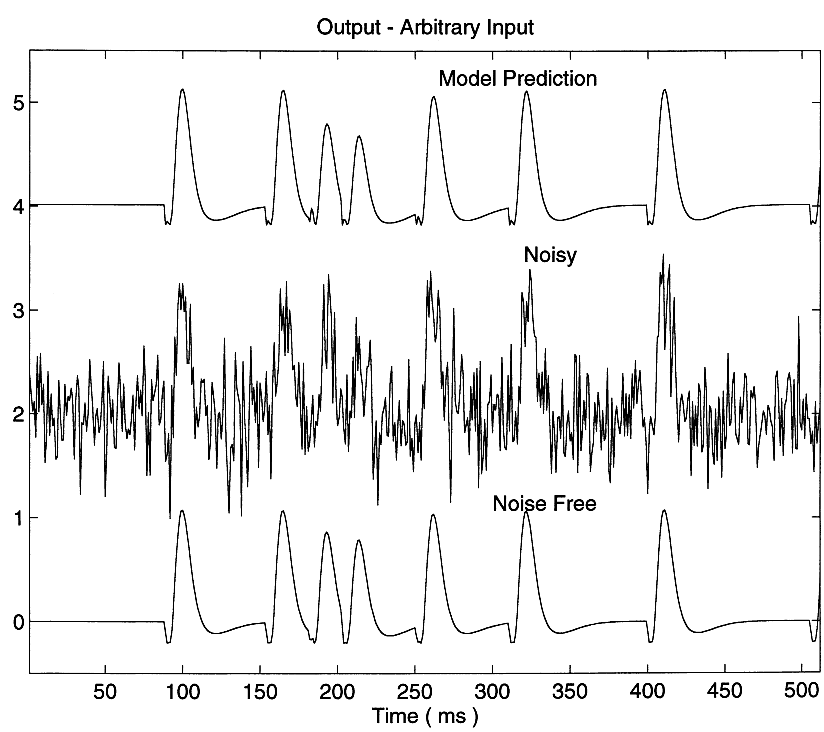

Fig. 4. The actual noisy output at SNR = 0 dB (middle trace) for an arbitrary (non-Poisson) point-process input and the model predicted output (top trace) are shown for a testing data record (not used for training). The noise-free output is also shown (bottom trace) to allow comparison with the model prediction and to demonstrate the robustness of this approach.

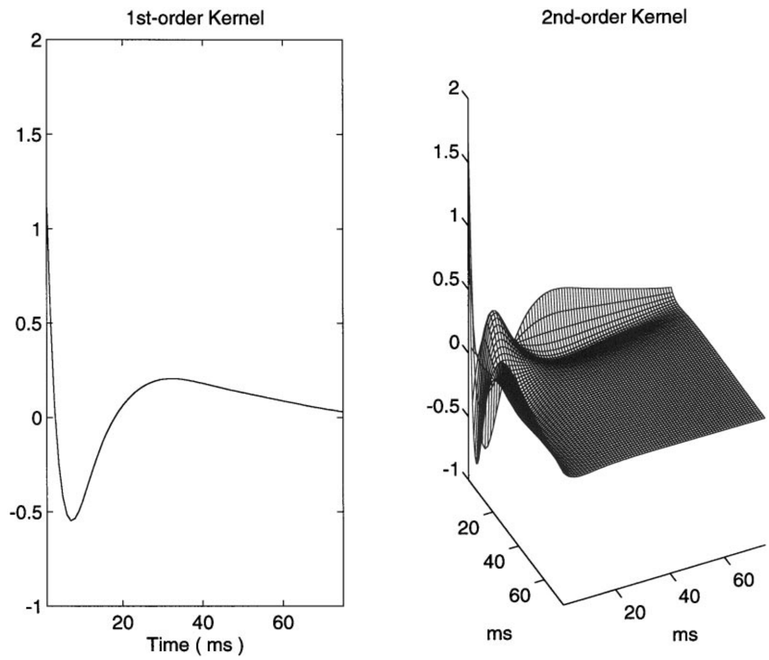

Fig. 5. Exact 1st and 2nd order kernels of the second simulated system.

Laguerre functions than necessary). When the LVN model parameters are overspecified, it was found that the method yields a precise model prediction and precise kernel estimates, but the individual estimates of the model parameters might vary because of redundancy. This important issue should be resolved in practice by means of successive trials in ascending numbers of parameters, so that by using the output prediction error as a guide, overspecification or underspecification of the model parameters can be avoided. A similar procedure is followed

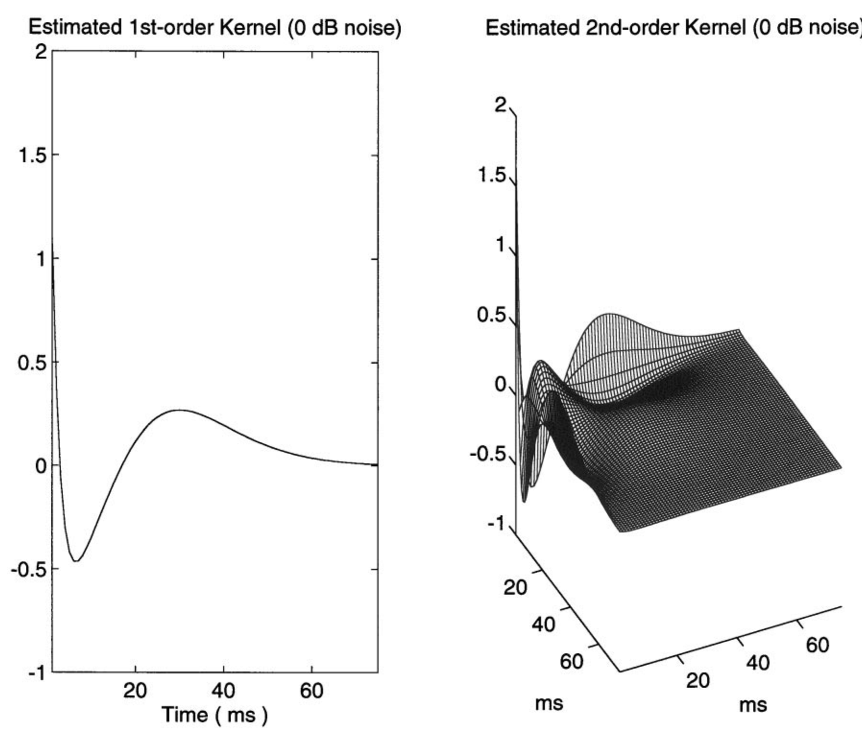

Fig. 6. The obtained 1st and 2nd order kernel estimates of the second simulated system for an arbitrary (non-Poisson) point-process input and independent Gaussian white noise added to the output resulting in the signal-to-noise ratio (SNR) of 0 db.

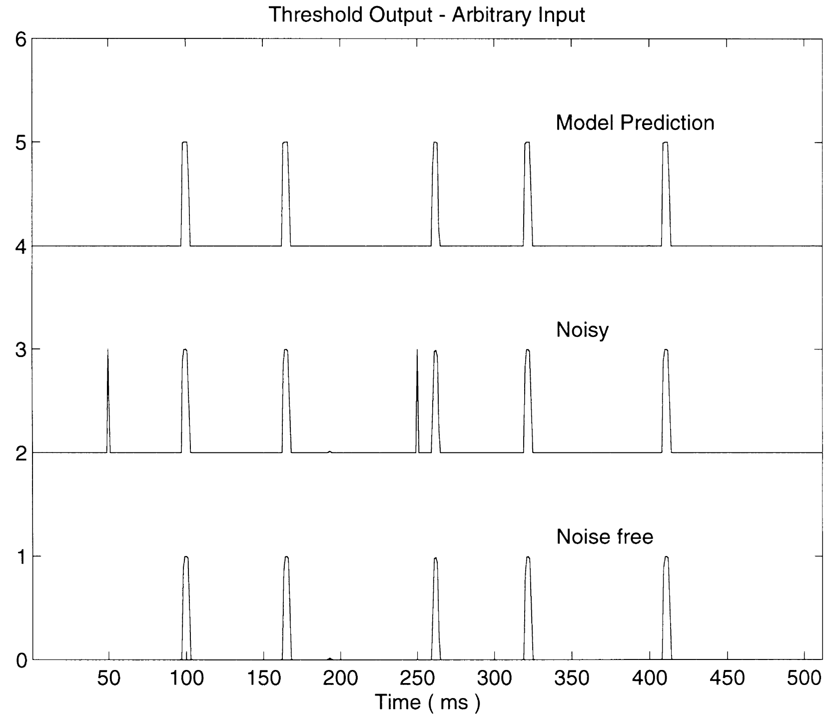

Fig. 7. Two independent (spurious) spikes are added to the system output when an arbitrary (non-Poisson) point-process is the system input. The resulting model prediction (top trace) along with the noisy output (middle trace) and the noise-free output (bottom trace) are shown for a testing data record (not used for training) and illustrate the robustness of this approach.

for the selection of the Laguerre parameter α. This procedure is demonstrated in Section 3.3 for a real neural system. It was further shown that the LVN modeling approach is robust in the presence of output-additive noise.

3.2. Point-process output

In order to examine the efficacy of this modeling approach in the case of point-process outputs, we append a threshold operator of the form given by Eq. (21) to the output of the first simulated system. The specific parameters of this threshold operator are trainable, as discussed at the end of Section 2 (Eqs. (23)–(25)). The composite static nonlinearity (i.e. the cascade of the polynomial activation function with the output threshold) can be approximated by a power series expansion (i.e. the system and the model are now of infinite Volterra order). This presents no practical problems, since the proposed method estimates the coefficients of the polynomial activation functions and the output threshold separately.

Obtained results of the model prediction in the noise-free cases (both Poisson and non-Poisson inputs) were again precise. Nonetheless, the resulting kernel estimates are affected by the presence of the threshold at the output, as anticipated by the theory (Marmarelis, Citron & Vivo, 1986). Note that there is an intrinsic scaling ambiguity between the threshold value and the coefficients of the polynomial activation functions, which however, does not alter the accuracy of the model prediction.

The effect of output-additive noise was examined here by adding independent (spurious) spikes to the system output (both for the training and the testing data-records). These spurious spikes can be viewed as point-process noise. The proposed method remains robust in the presence of such noise. For instance, when two spurious spikes are added to the eight spikes of the noise-free output of the testing data-record (25% contamination), the resulting model prediction ignores the spurious spikes in the noisy output, as shown in Fig. 7. Note that the output of the training data-record is also subject to 25% contamination by spurious spikes. This figure demonstrates the exceptional power of the proposed modeling approach for real neural systems subject to extraneous influences (spurious spikes).

3.3. A real neural system

To demonstrate the application of the proposed methodology to a real neural system, we consider the stimulus-response data collected from the dentate gyrus of a hippocampal slice preparation (Berger, Harty, Choi, Xie, Barrionuevo & Sclabassi, 1994). The stimulus pulses were delivered at random time intervals and the elicited excitatory post-synaptic potentials (EPSP) were recorded from the cell body layer of the dentate gyrus. A data-record of 15 spikes is analyzed (about 9200 data points) using the LVN with 7 Laguerre functions ($L = 6$) with

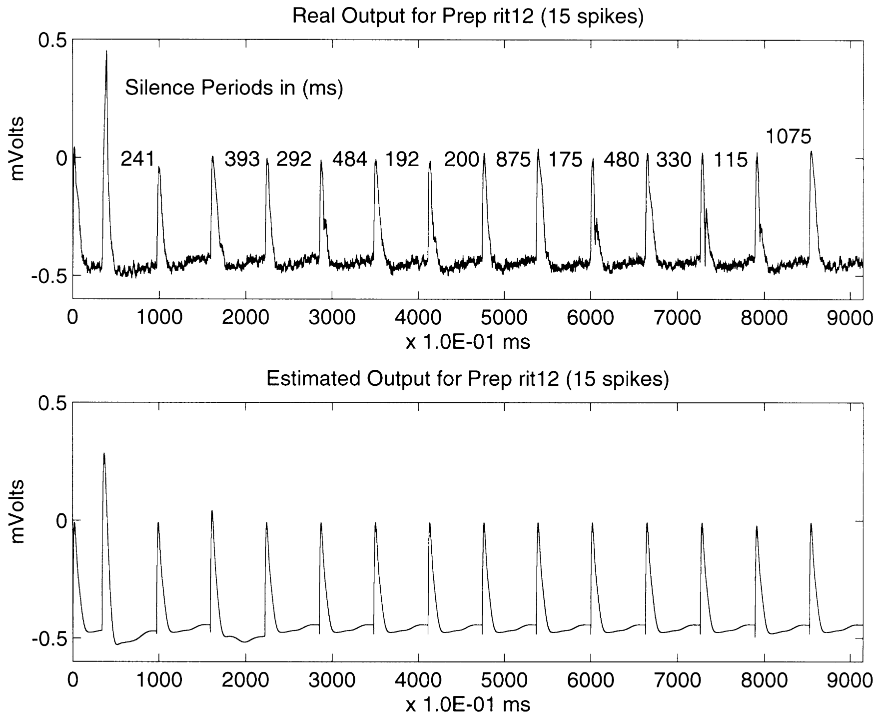

Fig. 8. The stimulus-response data collected from the dentate gyrus of a hippocampal slice preparation. The response to a stimulation record of 15 spikes is shown along with the LVNs model prediction. For clarity of presentation, we skip over the "silent intervals" between spikes and mark the length of each silent interval after the following output spike in number of sample points. When no number appears after a spike, there is no preceding silent period. The time denoted in the abscissa is not real time but "pseudo-time" because of the omission of the silent intervals.

$\alpha = 0.92$, two hidden units ($K = 2$) and third-degree polynomial activation functions ($Q = 3$). These model parameters were determined by successive trials, starting with low parameter values and increasing methodically each parameter until the MSE of the model prediction ceases being significantly reduced. Note that the selected LVN contains only 22 free parameters to be trained with the data.

Convergence was achieved within 100 iterations, and the resulting model prediction is shown in Fig. 8 along with the actual system response. Note that in order to facilitate the display of these data (sampling interval of 0.1 ms), we eliminate the "silent intervals" between EPSP's and mark the length of each silent interval after the following EPSP (in number of sample points). The result corroborates the efficacy of this approach, because of the good agreement between the model prediction and the actual system response in terms of the waveform and size of the EPSP (except for the high frequency noise evident in the response recording). The obtained first and second order kernels are shown in Fig. 9. These kernel estimates were consistent for different data segments and experimental preparations, attesting to the validity of the obtained LVN model. It is also worth noting that when the first ten EPSPs were used for training, the resulting LVN was able to predict the following five EPSPs. This demonstrates the generalization capability of the LVN.

4. Conclusions

A practicable approach to the modeling problem of nonlinear neural systems with arbitrary point-process inputs is proposed that employ a novel network architecture and adaptive estimation techniques (i.e. training of the network parameters by error back-propagation algorithms). The novelty of the network, termed the LVN is in utilizing a Laguerre filter bank for pre-processing the input data and using trainable polynomial activation functions in the single hidden layer. The LVN is also adapted to point-process outputs by appending a trainable threshold operator at the output.

It is shown with computer-simulated examples and real neural data that the proposed LVN methodology is capable of yielding accurate nonlinear models for both continuous and point-process outputs in response to arbitrary point-process inputs (so long as the latter do not attain a very restrictive deterministic form). This waives the restrictive requirement of Poisson input processes for kernel estimation via the CCT. The method also is shown to be robust in the presence of both continuous and point-process output-additive noise.

The proposed nonlinear modeling methodology offers an effective tool for the study of neural systems under conditions of spontaneous activity that cannot be studied with current nonlinear methods (since the latter require Poisson

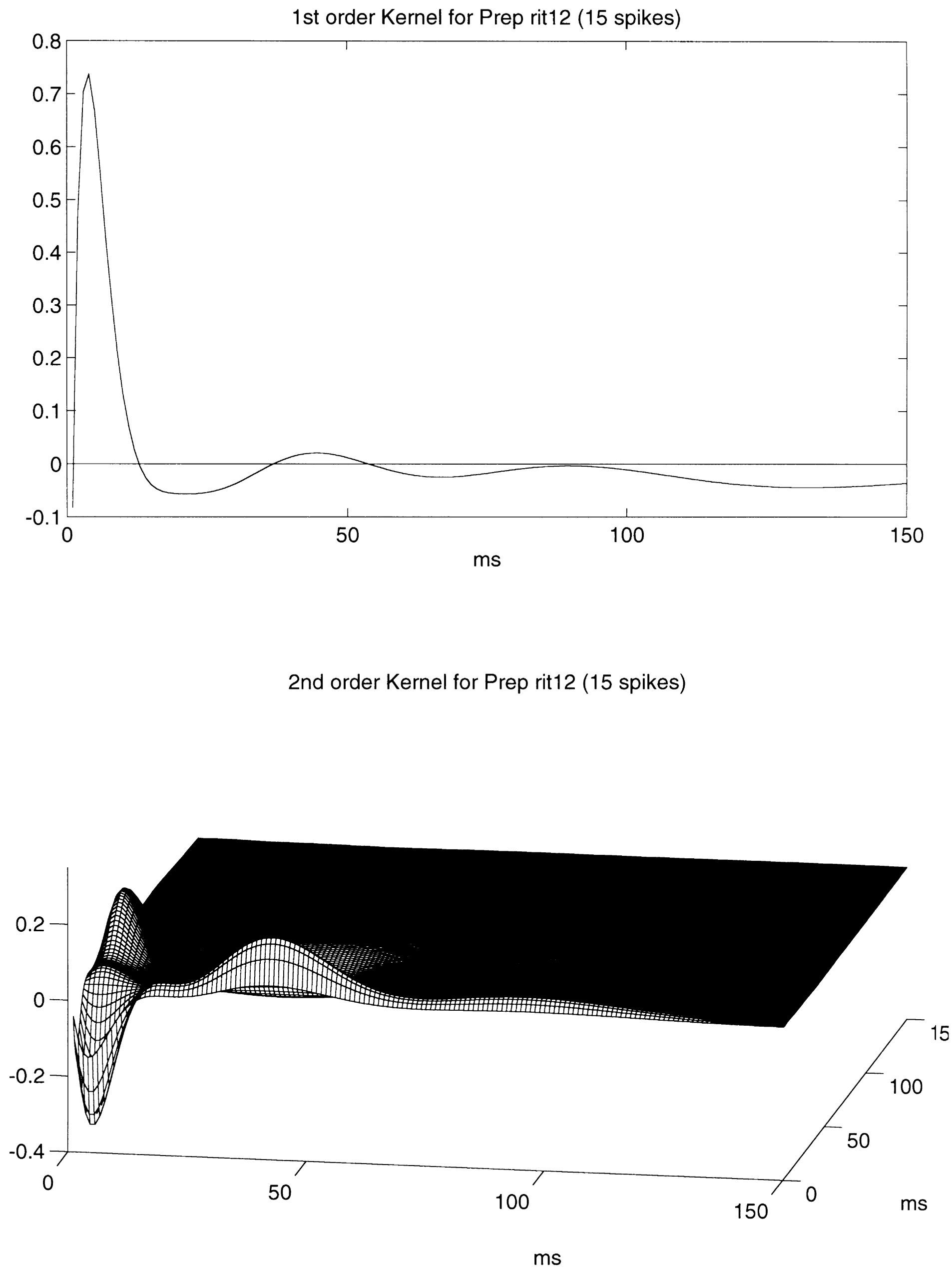

Fig. 9. The obtained 1st and 2nd order kernel estimates from the dentate gyrus of a hippocampal slice preparation using the LVN with 15 input–output spikes.

inputs). It can be extended to the case of multiple inputs and multiple outputs as to allow analysis of the data from multi-site recordings of neuronal ensembles.

Acknowledgements

This work was supported by NIH grants RR-01861 awarded to the Biomedical Simulations Resource at the University of Southern California from the National Center for Research Resources, MH-51722 and MH-00343 from the National Institute of Mental Health, and ONR contract No. N00014-94-1-0568.

References

Barrett, J. F. (1963). The use of functionals in the analysis of nonlinear physical systems. *Journal of Electronic Control*, *15*, 567–615.

Berger, T. W., Robinson, G. B., Port, R. L., & Sclabassi, R. J. (1987). Nonlinear systems analysis of the functional properties of the hippocampal formation. In V. Z. Marmarelis (Ed.), (pp. 73–103). *Advanced methods of physiological systems modeling*, 1. Los Angeles: Biomedical Simulation Resource.

Berger, T. W., Harty, T. P., Choi, C., Xie, X., Barrionuevo, G., & Sclabassi, R. J. (1994). Experimental basis for an input/output model of the hippocampus. In V. Z. Marmarelis (Ed.), (pp. 29–53). *Advanced methods of physiological system modeling*, 2. New York: Plenum.

Broome, P. W. (1965). Discrete orthonormal sequences. *Journal of the Association for Computing Machinery*, *12* (2), 00.

Chen, S., Billings, S. A., & Grant, P. M. (1990). Non-linear system identification using neural networks. *International Journal of Control*, *6*, 1191–1214.

Haykin, S. (1994). *Neural networks. A comprehensive foundation*, New York: Macmillan College Publishing Company.

Krausz, H. I. (1975). Identification of nonlinear systems using random impulse train inputs. *Biological Cybernetics*, *19*, 217–230.

Lee, Y. W., & Schetzen, M. (1965). Measurement of the Wiener kernels of a nonlinear system by crosscorrelation. *International Journal of Control*, *2*, 237–254.

Mäkilä, P. M. (1990). Approximation of stable systems by Laguerre filters. *Automatica*, *26* (2), 333–345.

Mäkilä, P. M. (1990). Laguerre series approximation of infinite dimensional systems. *Automatica*, *26* (6), 985–995.

Marmarelis, V. Z. (Ed.). (1987). Advanced methods of physiological system modeling, 1. Los Angeles, CA: University of Southern California, Biomedical Simulations Resource.

Marmarelis, V. Z. (Ed.). (1989). Advanced methods of physiological system modeling, 2. New York: Plenum Press.

Marmarelis, V. Z. (1993). Identification of nonlinear biological systems using Laguerre expansions of kernels. *Annals of Biomedical Engineering*, *21*, 573–589.

Marmarelis, V. Z. (Ed.). (1994). Advanced methods of physiological system modeling, 3. New York: Plenum Press.

Marmarelis, P. Z., & Marmarelis, V. Z. (1978). *Analysis of physiological systems: the white-noise approach*, New York: Plenum Press Russian translation: Mir Press, Moscow, 1990; Chinese translation: Academy of Sciences Press, Beijing, 1990.

Marmarelis, V. Z., & Orme, M. E. (1993). Modeling of neural systems by use of neuronal modes. *IEEE Transactions of Biomedical Engineering*, *40*, 1149–1158.

Marmarelis, V. Z., & Zhao, X. (1997). Volterra models and three-layer perceptrons. *IEEE Transactions of Neural Networks*, *8*, 1421–1433.

Marmarelis, V. Z., Citron, M. C., & Vivo, C. P. (1986). Minimum-order Weiner modeling of spike-output systems. *Biological Cybernetics*, *54*, 115–123.

Ogura, H., (1985). *Estimation of Wiener kernels of a nonlinear system and a fast algorithm using digital Laguerre filters*. 25th NIBB Conference, (pp. 14–62), Okazaki, Japan.

Rugh, W. J. (1981). *Nonlinear system theory: the Volterra/Wiener approach*, Baltimore, MD: Johns Hopkins University Press.

Schetzen, M. (1980). *The Volterra and Wiener theories of nonlinear systems*, Wiley: New York.

Volterra, V. (1930). *Theory of functionals and of integral and integro-differential equations*, New York: Dover.

Wahlberg, B. (1991). System identification using Laguerre-models. *IEEE Transaction of Automatic Control*, *36* (5), 00.

Wahlberg, B., & Mäkilä, P. M. (1995). On approximation of stable linear dynamical systems using Laguerre and Kautz functions. *Automatica*, *32* (5), 693–708.

Watanabe, A., & Stark, L. (1975). Kernel method for nonlinear analysis: identification of a biological control system. *Mathematical Biosciences*, *27*, 99–108.

Wiener, N. (1958). *Nonlinear problems in random theory*, New York: The Technology Press, MIT and Wiley.

Wray, J., & Green, G. G. R. (1994). Calculation of the Volterra kernels of nonlinear dynamic systems using an artificial neural network. *Biological Cybernetics*, *71*, 187–195.

Zurada, J. M. (1992). *Introduction to artificial neural systems*, St. Paul, MN: West Publishing Company.