Volterra Models and Three-Layer Perceptrons

Vasilis Z. Marmarelis, *Fellow, IEEE,* and Xiao Zhao, *Member, IEEE*

***Abstract*—This paper proposes the use of a class of feedforward artificial neural networks with polynomial activation functions (distinct for each hidden unit) for practical modeling of *high-order* Volterra systems. Discrete-time Volterra models (DVM's) are often used in the study of nonlinear physical and physiological systems using stimulus-response data. However, their practical use has been hindered by computational limitations that confine them to low-order nonlinearities (i.e., only estimation of low-order kernels is practically feasible). Since three-layer perceptrons (TLP's) can be used to represent input–output nonlinear mappings of arbitrary order, this paper explores the basic relations between DVM and TLP with tapped-delay inputs in the context of nonlinear system modeling. A variant of TLP with polynomial activation functions—termed "separable Volterra networks" (SVN's)—is found particularly useful in deriving explicit relations with DVM and in obtaining practicable models of highly nonlinear systems from stimulus-response data. The conditions under which the two approaches yield equivalent representations of the input–output relation are explored, and the feasibility of DVM estimation via equivalent SVN training using backpropagation is demonstrated by computer-simulated examples and compared with results from the Laguerre expansion technique (LET). The use of SVN models allows practicable modeling of high-order nonlinear systems, thus removing the main practical limitation of the DVM approach.**

***Index Terms*— Laguerre kernel expansion, nonlinear system modeling, polynomial activation functions, separable Volterra network, three-layer perceptrons, Volterra kernels, Volterra models.**

I. Introduction

THE Volterra approach to nonlinear system modeling has been used extensively in studies of physiological (and especially neural) systems for the last 25 years, following the customary cycle of exciting advances and confounding setbacks (for partial review, see [21]–[24]). On the other hand, feedforward artificial neural networks, and three-layer perceptrons (TLP's) in particular, have emerged in recent years as a promising approach to nonlinear mapping/modeling of input–output data (see, for instance, [13], [18], [31], and [32]). The rising interest in applications of these two approaches to nonlinear system modeling motivates this comparative study that seeks possible cross-enhancements from their combined use.

Manuscript received August 20, 1996; revised December 10, 1996 and August 9, 1997. This work was supported by Grant RR-01861 awarded to the Biomedical Simulations Resource at the University of Southern California from the National Center for Research Resources of the National Institutes of Health.

V. Z. Marmarelis is with the Department of Biomedical Engineering, University of Southern California, Los Angeles, CA 90089-1451 USA.

X. Zhao was with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089-1451 USA. He is now with the Biosciences and Bioengineering Division of the Southwest Research Institute in San Antonio, TX

Publisher Item Identifier S 1045-9227(97)08094-6.

Specifically, the study of high-order nonlinear physiological systems using discrete-time Volterra models (DVM's) is impeded by computational limitations in estimating high-order kernels. This problem may be mitigated by training equivalent TLP models with the available experimental data and seeking indirect estimation of high-order Volterra models via TLP with polynomial activation functions. Note that the latter are distinct for each hidden unit (i.e., have different coefficients), thus not contradicting previous results on the necessity of nonpolynomial activation functions with fixed form across all hidden units [2], [19]. On the other hand, applications of TLP can benefit from methodological guidance in selecting the appropriate network architecture (e.g., the number or type of hidden units—a matter critical for determining the efficacy of the training process and the predictive ability of the model) and from enhancements in scientific interpretation of the obtained results, based on equivalent DVM estimated from the same data.

The relationship between Volterra models (Volterra series) and feedforward multilayer neural networks has been previously examined in a rudimentary fashion [6], [12], and methods have been suggested for the indirect estimation of Volterra kernels if an equivalent TLP with sigmoidal or polynomial activation functions can be successfully trained [24], [38]. Chen and Manry have employed "polynomial basis functions" to model multilayer perceptrons and suggested that the resulting neural network is "isomorphic to conventional polynomial discriminant classifiers or Volterra filters" [4]. Specht has examined a polynomial adaline architecture for classification tasks [35]. Sandberg has given a general mathematical proof of a relevant aproximation theorem [33]. Polynomial perceptron architectures have been explored in the problem of communication channel equalization [3] and cochannel interference suppression [39], where the polynomial perceptron is defined as employing a full Volterra series expression in cascade with a sigmoidal activation function—an architecture far less parsimonious than using polynomial activation functions in a TLP (distinct for each hidden unit), as suggested in this paper. Volterra approximations of perceptrons for nonlinear noise filtering and beamforming have also been explored empirically [15]. Of particular theoretical and methodological interest is the relation between artificial neural networks and the generalized Fock space framework for nonlinear system modeling [8], [9], as well as the associated optimal interpolative nets [10], [34]. Finally, the established concept of "polynomial threshold gates," as applied to Boolean (switching) functions [13], is affine to—but distinct from—the concept of polynomial activation functions in feedforward neural networks.

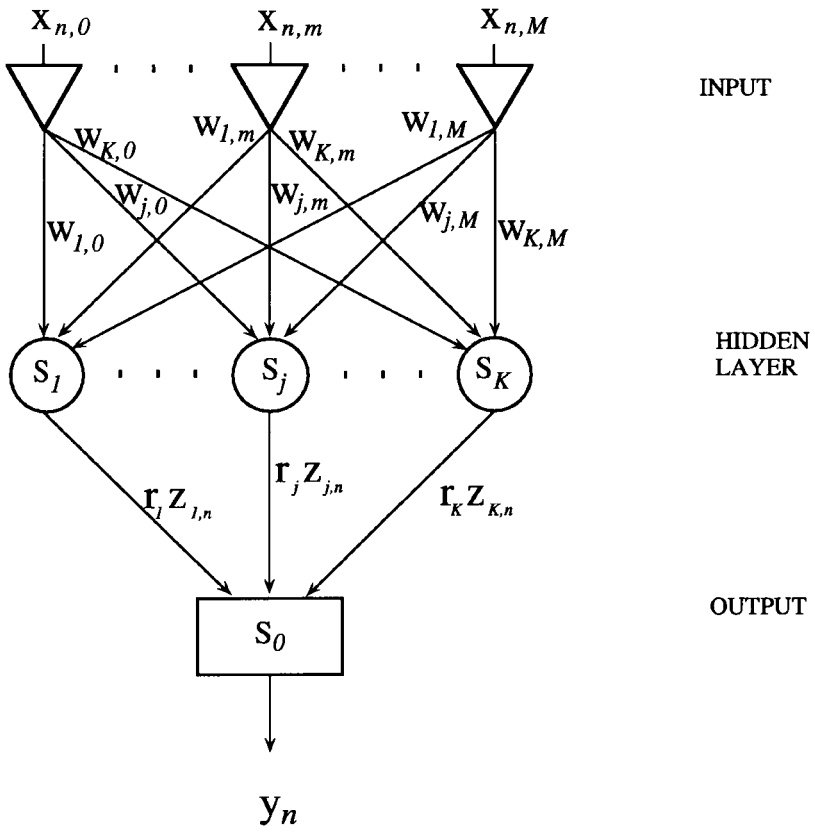


Fig. 1. Schematic diagram of the single-output three-layer perceptron (TLP) where the input vector represents the $(M+1)$-point input epoch at each discrete-time n. The jth hidden unit performs a sigmoidal transformation $S_j(\cdot)$ on the weighted sum of the input values with an offset θ_j. The output unit may perform a sigmoidal or linear transformation $S_0(\cdot)$ on the weighted sum of the outputs of the hidden units.

This paper examines the fundamental relations between DVM and TLP with tapped-delay input, and focuses on their cooperative use for practical modeling of nonlinear dynamic systems from stimulus-response sampled data. Both model types are shown to be able to represent nonlinear input–output mappings (systems), thus according them equal distinction as "universal approximators." Of particular interest is the use of distinct polynomial activation functions in the hidden units of TLP architectures that achieve modeling efficiencies and facilitate comparisons with DVM. Sections II and III review the basics of the TLP and DVM approaches, respectively. Section IV compares the two approaches, introduces the "separable Volterra networks" and discusses their equivalence conditions. Section V examines the relative efficacy of these approaches in modeling Volterra systems through computer simulated examples, where the Laguerre expansion technique (LET) is employed for DVM kernel estimation [26].

II. Three-Layer Perceptron with Single Output

The basic class of single-output TLP depicted in Fig. 1, implements a nonlinear mapping of the input epoch, represented by the vector $\underline{x}_n^T = [x_{n,0}\, x_{n,1} \cdots x_{n,M}]$, on the output scalar y_n at each time n. Since this study is concerned with input data that are ordered in discrete time sequence, we consider a tapped-delay input, where $x_{n,m} \equiv x(n-m)$ for each time index n. The case of a single output is considered in order to conform with the formalism of the Volterra expansion of a single-output system.

Each hidden unit of the TLP performs a nonlinear transformation of a weighted sum of the respective inputs for each n, using the "activation function" $S(\cdot)$. A sigmoidal or "squashing" function is traditionally used for this purpose. However, other functions can be used as well (e.g., polynomial, sinusoidal, Gaussian etc., or combinations thereof) depending on the objectives of a particular application. Thus, the output of the jth hidden unit $(j = 1, \cdots, K)$ for each n is

$$z_{j,n} = S_j(u_{j,n}) \tag{1}$$

where

$$u_{j,n} = \sum_{m=0}^{M} w_{j,m} x_{n,m}. \tag{2}$$

Clearly, for a tapped-delay network, $u_{j,n}$ is the convolution of the input signal with a finite impulse response $\{w_{j,m}\}$. If a sigmoidal activation function is used, then another free parameter, θ_j, is introduced as the characteristic "threshold" or "offset" of the jth unit. For instance, the "logistic" function

$$S_j(u_{j,n}) = \frac{1}{1 + \exp[-\lambda(u_{j,n} - \theta_j)]} \tag{3}$$

is a commonly used sigmoidal activation function. Note that, in addition to the offset θ_j, the exponent contains another

parameter λ, which is however fixed—i.e., it is not estimated from the data but is specified by the user. The parameter λ determines the transition slope from level 0 to level 1, and may affect the stability and convergence of the back-propagation training algorithm. As λ increases the sigmoidal transformation tends to a "hard threshold." Various other sigmoidal functions have been used (e.g., $\tanh, \arctan$ etc.) in the TLP literature.

For the output unit, we have

$$y_n = S_o\left\{\sum_{j=1}^{K} r_j z_{j,n}\right\}. \tag{4}$$

In order to simplify the comparison between this TLP network and the DVM that seeks to perform the same input–output mapping, we will consider the case of a linear output unit

$$y_n = \sum_{j=1}^{K} r_j z_{j,n}. \tag{5}$$

Note that many other classes of feedforward neural networks have been explored in the literature (e.g., having multiple hidden layers, nonsigmoidal activation functions, nondeterministic weights, bilinear weighted sums, units with intrinsic dynamics, etc.). It is critical to note that the use of nonsigmoidal activation functions may offer significant methodological advantages and yield modeling efficiencies (as elaborated in Sections IV and V).

In addition to feedforward neural networks, architectures with lateral connections between same-layer units or feedback connections between different layers (recurrent networks) have been explored in the neural network literature and are suitable for certain applications. However, they result in far more complicated relations with the DVM that impede lucid comparisons. Hence, the scope of this study is limited to an explicit comparison between this relatively simple class of TLP networks and the DVM, since they represent two fundamental and general model forms for nonlinear input–output mappings of time-series data.

III. Discrete-Time Volterra Models

The DVM is valid for all continuous, causal, nonlinear, time-invariant systems/mappings with finite memory M

$$y(n) = \sum_{i} \sum_{m_1,\cdots,m_i=0}^{M} k_i(m_1,\cdots,m_i)x(n-m_1) \cdots x(n-m_i) \tag{6}$$

where $x(n)$ denotes the input data sequence and $y(n)$ the output data sequence. The kernel functions $\{k_i\}$ describe the nonlinear dynamics of the system (i.e., fully characterize the nonlinear input–output mapping) and they are symmetric (i.e., invariant to any permutation of their arguments). The input–output relation described by the DVM of (6) is functionally equivalent to the mapping effected by the TLP of Fig. 1.

The DVM can be viewed as a multivariate power series (multinomial, if of finite order) expansion of a nonlinear function

$$y(n) = F(x_{n,0}, x_{n,1}, \cdots, x_{n,M}) \tag{7}$$

where the argument of $F(\cdot)$ corresponds to the input epoch values at each time n, i.e., $x_{n,m} \equiv x(n-m)$. The ith functional term of (6) is an i-tuple convolution involving i time-shifted versions of the input epoch over the interval $[n, n-M]$ and the ith-order kernel k_i. This hierarchical structure defines a canonical representation of stable nonlinear causal systems (mapping operators), where the ith term represents the ith-order nonlinearities. Causality implies that future input values do not affect the present value of the output. Stability implies absolute summability of the Volterra kernels and convergence of the corresponding series of uniform bounds [22].

In this formulation, the class of linear systems is represented simply by the first-order term (the first-order kernel is the familiar "impulse response function") and the nonlinear system dynamics are explicitly represented by the corresponding high-order kernels. The degree of system nonlinearity determines the required number of kernels for a model of adequate predictive capability, subject to practical computational considerations.

This modeling approach has been used extensively in studies of physiological systems (especially neural systems) over the last 25 years. Following Wiener's pioneering ideas, its use has been combined with approximate white-noise stimuli (e.g., band-limited white-noise, binary, and ternary pseudorandom signals, etc.) in order to secure exhaustive testing of the system and facilitate the estimation of high-order kernels [37].

Extensive studies have explored the limits of applicability and efficient implementation of this approach, leading to successful applications to low-order nonlinear systems (up to third order). This modeling approach has been extended to the cases of multiple inputs and multiple outputs (including spatiotemporal inputs in the visual system), point-process inputs/outputs (suitable for neuronal systems receiving/generating action potentials) and time-varying systems often encountered in physiology. The main limitations of this approach are the practical inability to extend kernel estimation to orders higher than third (due to increasing dimensionality of kernel representation) and the strict input requirements (i.e., approximate white noise) for unbiased kernel estimation when truncated models are obtained. These limitations provide the motivation for seeking the cooperative use of TLP models.

If we consider an expansion of the Volterra kernels on a complete basis $\{b_j(m)\}$ of L basis functions defined over the system memory $[0, M]$, then the DVM of (6) becomes

$$y(n) = c_o + \sum_{j} c_1(j)v_j(n) + \sum_{j_1}\sum_{j_2} c_2(j_1,j_2)v_{j_1}(n)v_{j_2}(n) + \cdots \tag{8}$$

where

$$v_j(n) = \sum_{m=0}^{M} b_j(m)x(n-m) \tag{9}$$

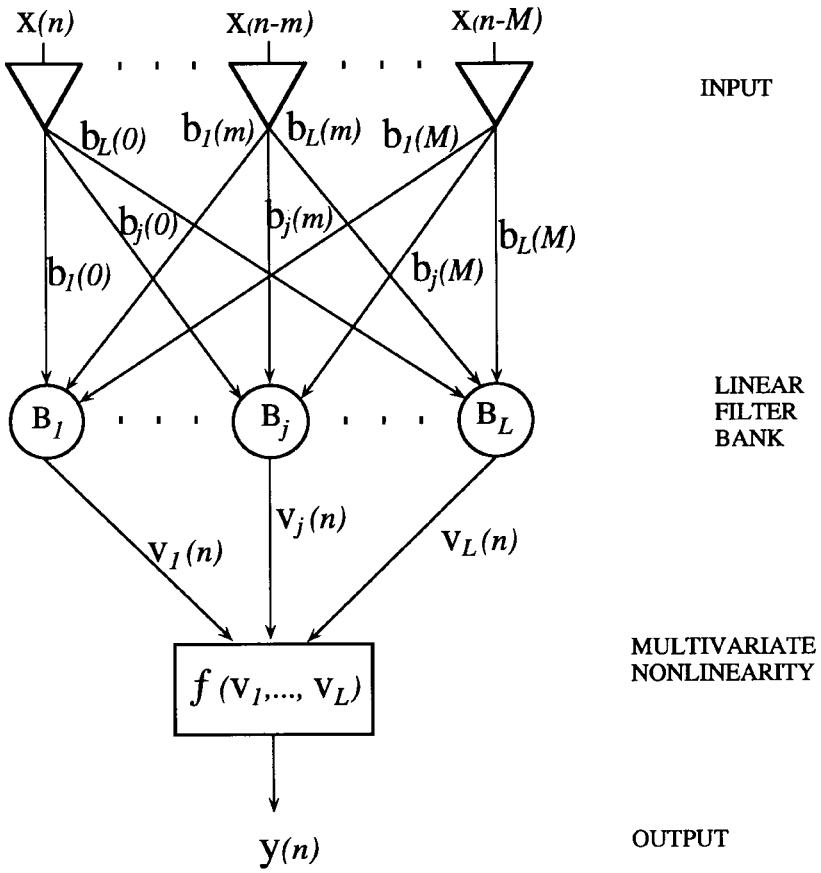


Fig. 2. Schematic diagram of themodified Volterra model (MVM) defined by (9) and (12). The middle-layer units $\{B_j\}$ form a linear filter-bank thatspans the system dynamics and generates the signals $\{v_j(n)\}$. The latter are the inputs of the multivariate nonlinear function $f(\cdot)$ that represents the nonlinearities of the Volterra model.

and $c_1, c_2, \cdots$ are the kernel expansion coefficients ($c_o = k_o$). For instance,

$$k_1(m) = \sum_{j=1}^{L} c_1(j) b_j(m) \tag{10}$$

$$k_2(m_1, m_2) = \sum_{j_1=1}^{L} \sum_{j_2=1}^{L} c_2(j_1, j_2) b_{j_1}(m_1) b_{j_2}(m_2) \tag{11}$$

are the expansions for the first- and second-order kernels satisfying the weak condition of square summability over $[0, M]$. These expansions are extended to all kernels present in the system.

Note that the variable $v_j(n)$ is a weighted sum of the input epoch values (i.e., discrete convolution), akin to the variable $u_{j,n}$ of (2). This is an important common feature of the two canonical representations (TLP and DVM). It is critical to note that, the number L of basis functions $\{b_j\}$ required for adequate approximation of the kernels can be made much smaller than M by choosing the proper basis $\{b_j\}$ in a given application. This leads to more compact models due to reduction of the dimensionality of the $(M+1)$-dimensional function $F(\cdot)$ in (7), yielding the L-dimensional output function

$$y = f(v_1, v_2, \cdots, v_L) \tag{12}$$

which is equivalent to the expression of (8). The expression of (8) can be viewed as a multivariate (Taylor) expansion of an analytic nonlinear function $f(\cdot)$ or as a multinomial approximation of a nonanalytic function $f(\cdot)$, termed the modified Volterra model (MVM).

The MVM corresponds to the functional diagram (network) of Fig. 2, where the middle-layer units $\{B_j\}$ form a linear filter-bank with impulse response functions $\{b_j(m)\}$ performing discrete-time convolutions with the input data, as indicated by (9). This filter-bank may be formed by an arbitrary (general) set of basis functions or it may be "customized" for the particular dynamic characteristics of a given system (mapping) to achieve compactness and computational efficiency (possibly using an adaptive approximation process). This customized basis can be constructed from estimates obtained from a general basis (e.g., the Laguerre set for causal systems). One such method has been recently proposed that uses eigen-decomposition of an estimated second-order Volterra model to select the functions $\{b_j(m)\}$ as the "principal dynamic modes" of the nonlinear system [27], [28]. The pursuit of parsimony motivates the search for the most efficient basis $\{b_j\}$, which may or may not be orthogonal.

This type of decomposition of a nonlinear system was first proposed (in continuous time) by Wiener in connection with a complete orthonormal basis $\{b_j\}$ and a Gaussian white

noise input [37]. In Wiener's formulation, the variables $\{v_j\}$ become independent Gaussian processes and the nonlinearity $f(\cdot)$ is further expanded on a Hermite orthonormal basis for the purpose of nonlinear system identification. The Wiener formulation may not be a practical or desirable option in a given application; nonetheless, it represents a powerful conceptual framework for nonlinear modeling, demonstrating the ability of white-noise inputs to extract sufficient information from the system for constructing nonlinear models. Consequently, white-noise stimuli (whenever available) constitute an effective ensemble of inputs for system modeling or network training.

Of particular interest in modeling studies of real neural systems that generate action potentials is the case of spike-output models. This case has been studied by appending a (hard) threshold-trigger operator to the output of the previously discussed MVM model. This formulation leads to exact models by defining "trigger regions" (TR's) of the system as the locus of points $(v_1, v_2, \cdots, v_L)$ in the L-dimensional space which correspond to the appearance of an output spike [25], [28], [29].

The introduction of a hard-threshold θ at the output of the nonlinear function $f(\cdot)$ implies that the aforementioned TR's are demarcated by the "trigger boundaries" (TB's) defined by the equation

$$f(v_1, v_2, \cdots, v_L) = \theta. \tag{13}$$

These TB's correspond to the "decision boundaries" encountered in TLP applications with binary outputs, and may take any form depending on the function $f(\cdot)$, i.e., they are not limited to piecewise rectilinear forms dictated by TLP configurations. Illustrative examples of this important comparison are given in the following section.

IV. Comparison Between TLP and DVM

To facilitate the comparison between the two models/networks of Figs. 1 and 2, we first assume that the output unit of the TLP is linear [see (5)]. Then, using the Taylor series expansion of each sigmoidal function about its offset value θ_j

$$S_j(u_{j,n}) = \sum_{i=0}^{\infty} \alpha_i(\theta_j) u_{j,n}^i \tag{14}$$

we can express the output as

$$y_n = \sum_{j=1}^{K} r_j \sum_{i=0}^{\infty} \alpha_i(\theta_j) \times \sum_{m_1=0}^{M} \cdots \sum_{m_i=0}^{M} w_{j,m_1} \cdots w_{j,m_i} x_{n,m_1} \cdots x_{n,m_i} \tag{15}$$

which has the form of a Volterra series expansion, where $y_n = y(n), x_{n,m} = x(n-m)$ and

$$k_i(m_1, \cdots, m_i) = \sum_{j=1}^{K} r_j \alpha_i(\theta_j) w_{j,m_1} \cdots w_{j,m_i}. \tag{16}$$

The Taylor expansion coefficients $\{\alpha_i(\theta_j)\}$ depend on the offsets $\{\theta_j\}$ and are characteristic of the sigmoidal or any other analytic activation function [38]. Likewise, a finite polynomial approximation can be obtained for any continuous activation function. Thus, (16) can be used to evaluate the Volterra kernels of a TLP model of a system. We will return to this issue in the following section.

In searching for the equivalence conditions between TLP and MVM, we note that in both cases "hidden" variables are used (i.e., u and v, respectively) that are formed by linear combinations (convolutions) of the input vector values according to (2) and (9), respectively. Thus, the role of the filter-bank in Fig. 2 mirrors the role of the in-bound TLP weights in Fig. 1. If a filter-bank $\{b_j(m)\}$ can be found such that the resulting multivariate output function $f(v_1, \cdots, v_L)$ can be expressed as a linear superposition of sigmoidal univariate functions

$$f(v_1, \cdots, v_L) = \sum_{j=1}^{L} r_j S_j(v_j) \tag{17}$$

for some values of the parameters $\{r_j, \theta_j\}$, then the MVM and the TLP representations are equivalent. This equivalence condition can be broadened to cover activation functions other than sigmoidal (e.g., polynomial, which are directly compatible with the multinomial form of the MVM), in order to relax the conditions under which the continuous multivariate function $f(\cdot)$ can be represented (or approximated adequately) by a linear superposition of univariate functions.

This fundamental issue of representation of an arbitrary continuous multivariate function by the superposition of univariate functions was originally posed by Hilbert in 1902 (his so-called "13th problem") and has been addressed by Kolmogorov's representation theorem in 1957 [16] and its many elaborations (see, for instance, [7], [20], and [36]). This issue has regained importance with the increasing popularity of feedforward neural networks as "function approximators," especially since actual implementation of Kolmogorov's theorem leads to rather peculiar univariate functions [7]. The use of fixed activation functions (possibly nonsigmoidal) in multilayer perceptrons to obtain universal approximators has been recently studied by various investigators [1], [5], [11], [14]. Resolution of this issue with regard to nonlinear system modeling is achieved by reference to their canonical Volterra representation, as outlined below.

In the context of MVM, this issue concerns the representation of the output multivariate function $y = f(v_1, \cdots, v_L)$ by means of linear superposition of selected univariate functions $\{g_j(v_j)\}$ as

$$f(v_1, \cdots, v_L) = r_1 g_1(v_1) + \cdots + r_L g_L(v_L). \tag{18}$$

Note that, unlike (17), (18) allows for univariate continuous functions $\{g_j(\cdot)\}$ that have arbitrary forms (suitable for each application) leading to the network model form shown in Fig. 3. The latter is akin to the general "parallel cascade" models previously proposed for nonlinear time-invariant discrete-time systems [17], [30].

When the representation of (18) is possible, we can view the resulting model as a "generalized TLP" network with arbitrary activation functions that need not be sigmoidal. If we wish to facilitate comparisons with MVM, we can use polynomial

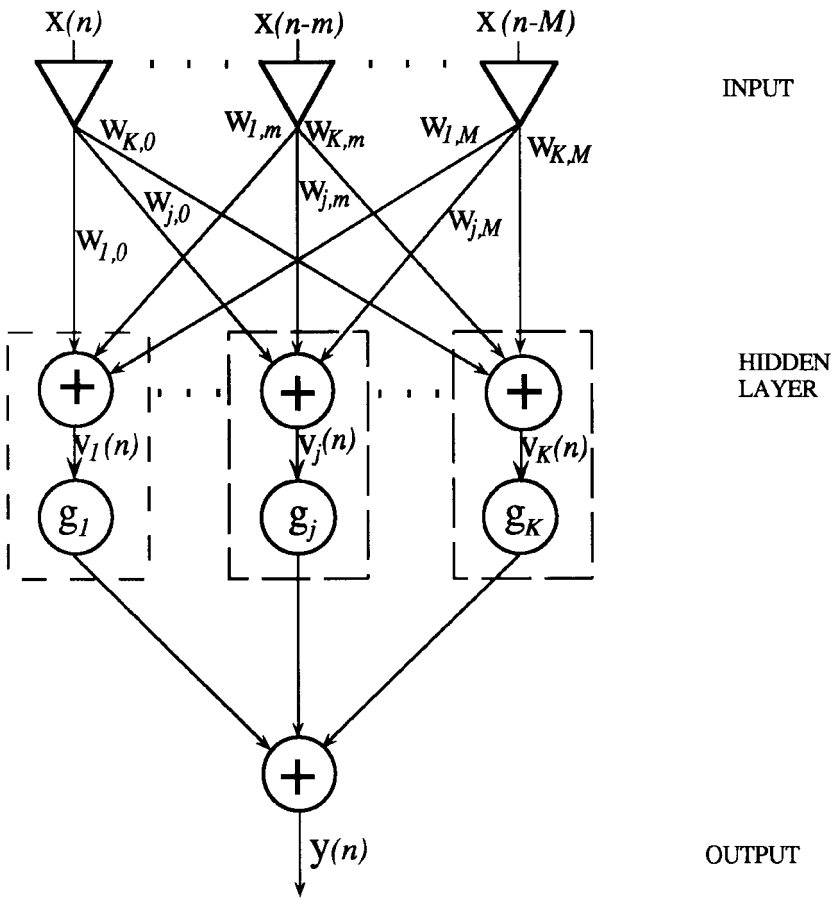


Fig. 3. Schematic diagram of the "separable Volterra network" (SVN) with *polynomial* activation functions $\{g_i\}$ in the hidden units. The output unit is a simple adder. This network configuration is compatible with Volterra models of nonlinear systems (mappings).

activation functions $\{g_i(v_i)\}$, leading to what we will term a "separable Volterra network" (SVN).

Illustrative comparisons between TLP and SVN are made easier when the sigmoidal functions $S(\cdot)$ of the TLP tend to a hard-threshold operator $T(\cdot)$, defining piecewise rectilinear TB's in the $\{u_j\}$ space on the basis of the equation

$$\sum_{j=1}^{K} r_j T(u_j) = \theta \tag{19}$$

where θ is the employed hard threshold at the output.

On the other hand, the use of a hard-threshold θ at the output of the MVM (or SVN) yields curvilinear TB's in the $\{v_i\}$ space, defined by the equation

$$f(v_1, \cdots, v_L) = \theta. \tag{20}$$

Since both the K-dimensional u-space and the L-dimensional v-space are defined by linear transformations of the input vectors, it is evident that the TB's defined by (19) and (20) cannot coincide unless $f(\cdot)$ is piecewise rectilinear. If $f(\cdot)$ is curvilinear, then we can achieve a satisfactory piecewise rectilinear approximation of the curvilinear TB by increasing the number K of rectilinear segments. Exact equivalence is achieved as K tends to infinity. This is illustrated below with a simple example.

Consider a Volterra system (mapping) that has two modes (b_1, b_2) with respective outputs (v_1, v_2) and the quadratic system nonlinearity: $f(v_1, v_2) = v_1^2 + v_2^2$, with output unit threshold: $\theta = 1$. Then (20) defines the circular TB: $v_1^2 + v_2^2 = 1$, in the (v_1, v_2) plane. For simplicity of demonstration, let us assume here that $b_1(m) = \delta(m)$ and $b_2(m) = \delta(m-1)$; which implies that $v_1(n) = x(n), v_2(n) = x(n-1)$, and an output spike occurs when $x^2(n) + x^2(n-1) \geq 1$, defining the circular TB shown in Fig. 4, with the "trigger region" found outside the circle. This system can be precisely modeled by a SVN having two hidden units (corresponding to the two modes) with second-degree polynomial activation functions and a unity threshold at the output (adder) unit. In order to approximate this circular TB by means of a TLP, we must use a large number K of hidden units which define different linear combinations of the two mode outputs and approximate the circular TB with K rectilinear segments over the domain defined by the training data set. An exact TLP model with the same predictive ability as this two-mode SVN can be obtained only when the number of hidden units tends to infinity.

As an illustration of this, we train a TLP with three hidden units using 500 datapoints generated by a uniform white-noise input that defines the square domain of values demarcated in Fig. 4 by dashed line. The resulting approximation is the triangle defined by the three rectilinear segments shown with dotted lines in Fig. 4. Considerable areas of "false positives"

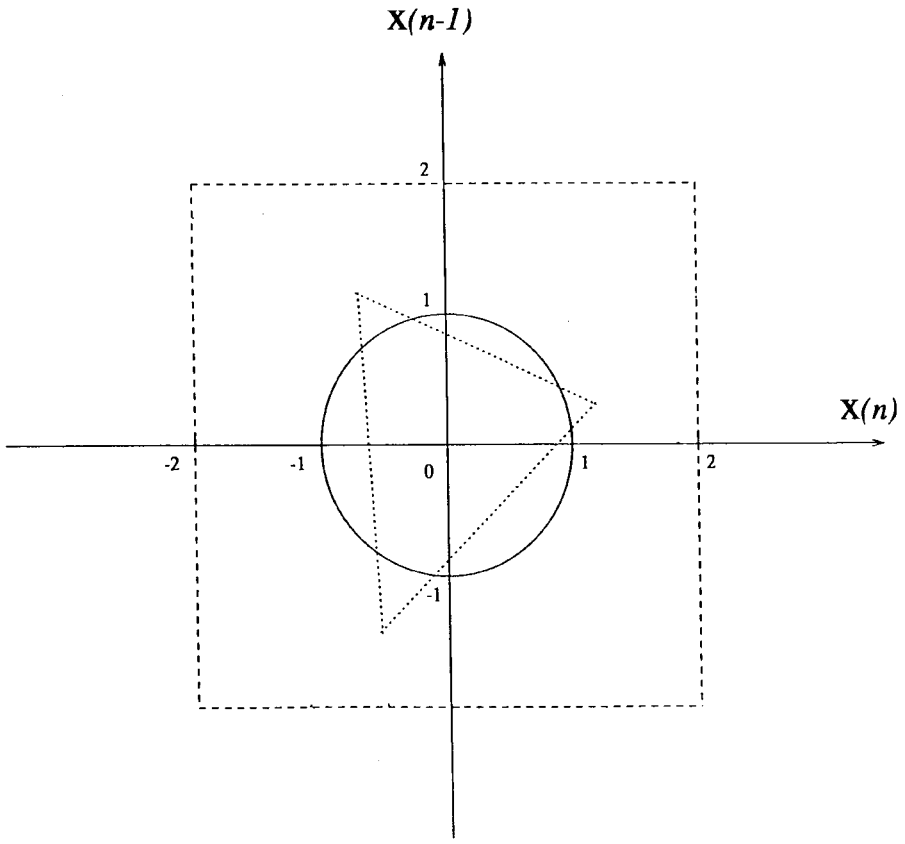


Fig. 4. Illustrative example of circular "trigger boundary" (solid line) being approximated by three-layer perceptron (TLP) with three hidden units defining the piecewise rectilinear (triangular) "trigger boundary" marked by the dotted lines. The training set is generated by 500 datapoints of uniform white noise input that defines the square domain demarcated by dashed lines. The piecewise rectilinear approximation improves with increasing number of hidden units of the TLP, assuming polygonal form and approaching asymptotically a precise representation. Nonetheless, a SVN with two hidden units (of second degree) yields a precise and parsimonious representation (model).

and "false negatives" are evident, which can be reduced only by increasing the number of hidden units and obtaining polygonal approximations of the ideal circular TB. The obtained TLP approximation depends on the specific training set and the initial parameter values—although fundamentally limited to a number of rectilinear segments equal to the number of employed hidden units.

Since many real nonlinear systems (physical or physiological) have been shown to be amenable to Volterra representations, it is reasonable to expect that the SVN formulation will yield more compact and precise models than the traditional TLP formulation. In addition, the SVN formulation is not practically limited to low-order nonlinearities (like the traditional Volterra modeling approach based on kernel estimation) thus allowing the obtainment of compact high-order nonlinear models with ordinary computational means.

This example demonstrates the potential benefits in model/network compactness and precision that may accrue from using the SVN configuration instead of the conventional TLP, whenever the "decision boundaries" (or TB's) are curvilinear.

Note that for SVN configurations, the output weights are set to unity (i.e., the output unit is a simple adder) without loss of generality, and the inbound weight vectors for each hidden unit are normalized to unity Euclidean norm in order to facilitate comparisons of the relative importance of different hidden units (as well as the relative importance of different polynomial terms) based on the absolute value of the coefficients of their polynomial activation functions. The SVN formulation also yields insight into the degree and form of system nonlinearities, as well as captures the input patterns that critically affect the output (akin to principal modes in a nonlinear context).

Another interesting comparison concerns the number of free parameters contained in the three types of models (TLP, MVM, SVN). If we use the same number of $(M+1)$ input units, then the total number of parameters for TLP with single-parameter sigmoidal functions is

$$N_{\mathrm{TLP}} = (M+3)K+1 \tag{21}$$

where K is the number of hidden units. In the case of MVM, if the highest power necessary for the adequate multinomial representation of the function $f(\cdot)$ is Q, then the total number of parameters is

$$N_{\mathrm{MVM}} = L(M+1) + \frac{(Q+L)!}{Q!L!} \tag{22}$$

where L is the number of employed modes (basis functions). Clearly, N_{MVM} depends critically on L and Q due to the factorials in (22). Note that the number of free parameters of

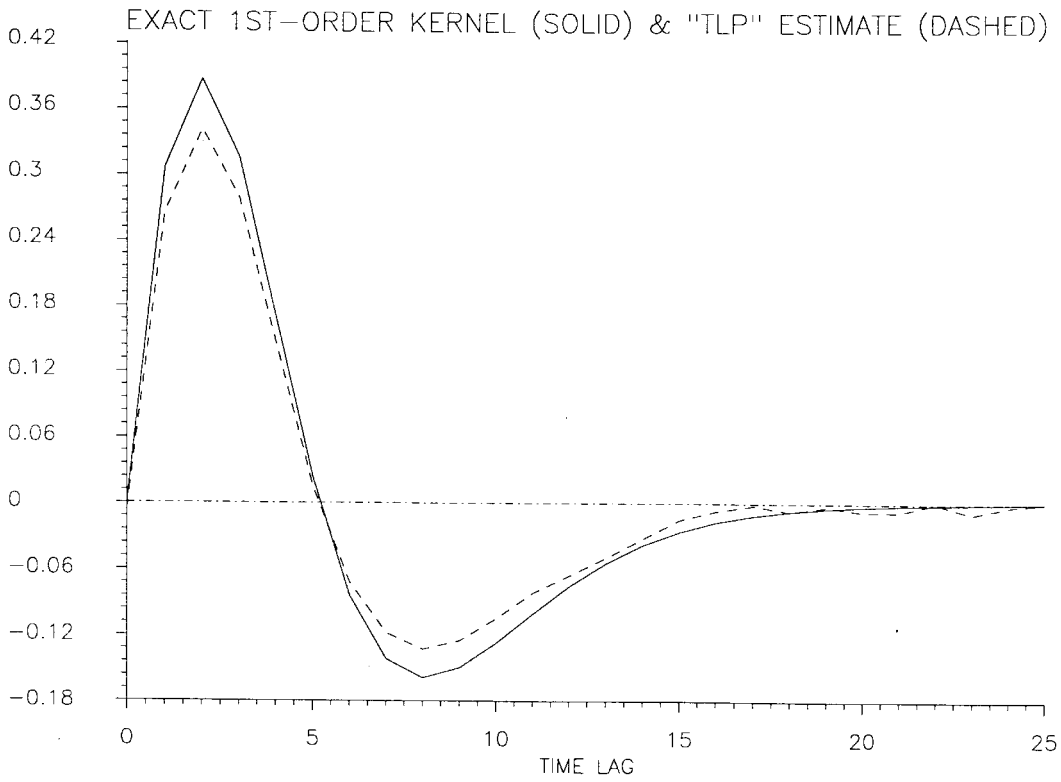


Fig. 5. The exact first-order kernel (solid line) and its estimate for the noise-free case using a TLP with four hidden units (dashed line). Note that the SVN and LET estimates are visually indistinguishable from the exact kernel in this case.

the original DVM is given by (22) for $L = M + 1$. Finally, in the case of SVN (of the same degree Q for all polynomial activation functions), the total number of parameters is

$$N_{\text{SVN}} = L(M + Q + 3) \tag{23}$$

and represents a compromise (in terms of parameterization) between TLP and MVM. We must keep in mind that K in (21) may have to be much larger than L in (22) and (23) in order to achieve the same model prediction accuracy. Thus, parsimony using SVN depends on our ability to determine the minimum number of modes (L) necessary for adequate prediction accuracy in a given application.

It is important to note that the type of available training data is also critical. If the available input data is rich in information content (i.e., approaching the case of quasiwhite noise signals) then the training of the network will be most efficient by exposing it to a diverse repertoire of possible inputs. However, if the training data is a limited subset of the entire input data space, then the system will be tested only partially and the results of network training will be limited in their ability to generalize.

V. Examples of Volterra System Modeling

As indicated previously, Volterra models can be used for a very broad class of nonlinear systems. However, applications of this modeling methodology to real systems have been hindered by the impracticality of estimating high-order kernels (corresponding to high-order nonlinearities). A solution to this important problem can be acheived by backpropagation training of high-order nonlinear models in the SVN or TLP form, that allow indirect evaluation of the system kernels from the obtained network parameters. In this section, we examine the relative efficacy of these kernel estimation methods and demonstrate the superior performance of the SVN formulation over the TLP approach for high-order Volterra system modeling.

It was shown earlier that equivalent Volterra kernels can be obtained from a TLP when the sigmoidal activation functions are expanded in a Taylor series as in (16) [38]. For the case of SVN with polynomial activation functions

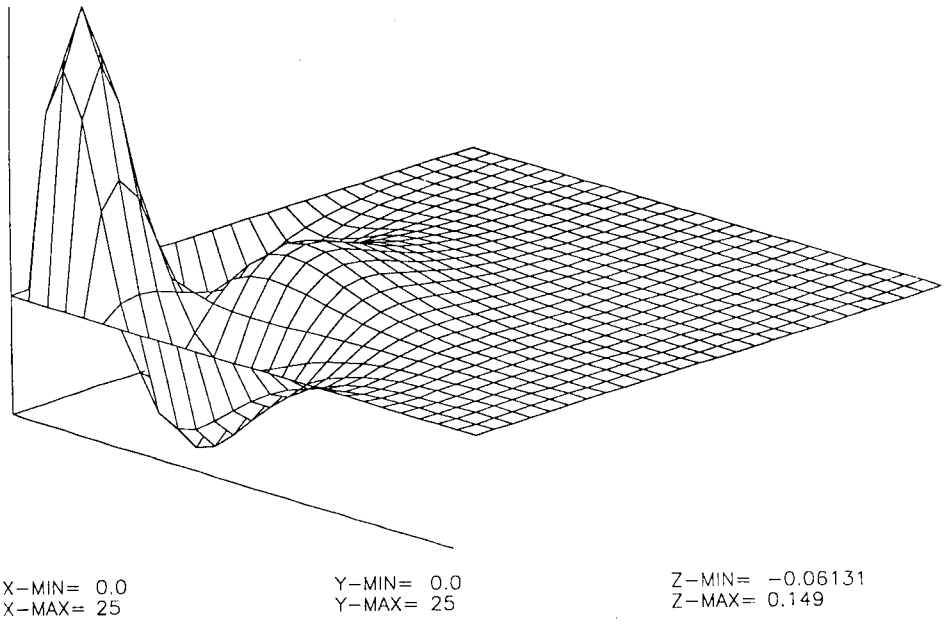
$$g_j(v_j) = \alpha_{0,j} + \alpha_{1,j} v_j + \cdots + \alpha_{i,j} v_j^i + \cdots \tag{24}$$

the expression for the ith-order Volterra kernel is slightly different

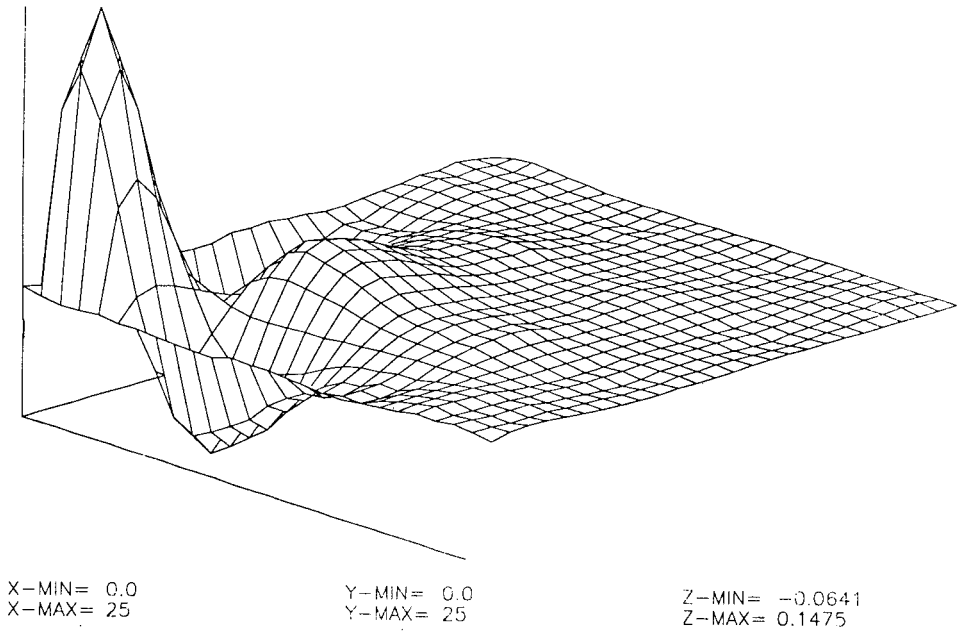
$$k_i(m_1, \cdots, m_i) = \sum_j \alpha_{i,j} w_{j,m_1} \cdots w_{j,m_i}. \tag{25}$$

Thus Volterra kernel estimation of any order can be accomplished by training a SVN with given input–output data and using (25) to reconstruct the kernels from the obtained weights $\{w_{j,m}\}$ and the coefficients $\{\alpha_{i,j}\}$ of the polynomial activation functions (all the unknown parameters obtained via error backpropagation). If the activation functions are nonpolynomial analytic or continuous functions, then the coefficients $\{\alpha_{i,j}\}$ correspond to a Taylor expansion or Weierstrass approximation, respectively.

As an illustrative example of the relative efficacy of these kernel estimation methods, we simulate a second-order



(a)



(b)

Fig. 6. (a) The exact second-order kernel which is indistinguishable from the SVN and LET estimates. (b) TheTLP estimate of the second-order kernel using four hidden units for the noise-free case. Increasing the number of TLP hidden units to eight did not yield significant improvement but added considerable computational burden.

Volterra system with memory $M = 25$ (having the first-order kernel shown in Fig. 5 in solid line and the second-order kernel shown in the top panel of Fig. 6) using a uniform white-noise input of 500 datapoints. We estimate the first- and second-order kernels of this system via TLP, SVN, and LET, which was recently introduced to improve kernel estimation over traditional methods by use of Laguerre expansions of the kernels and least-squares estimation of the expansion coefficients [26]. In this noise-free example, the LET and SVN approaches yield precise first- and second-order kernel estimates, although at considerably different computational cost (LET is about 20 times faster than SVN in this case). Note that LET requires five Laguerre functions in this example (i.e., 21 free parameters need be estimated) while SVN needs only one hidden unit with a second-degree activation function (resulting in 29 free parameters). Of course, the number of required Laguerre functions and hidden units may vary from application to application.

As expected, the TLP requires more free parameters in this example (i.e., more hidden units) and its predictive accuracy

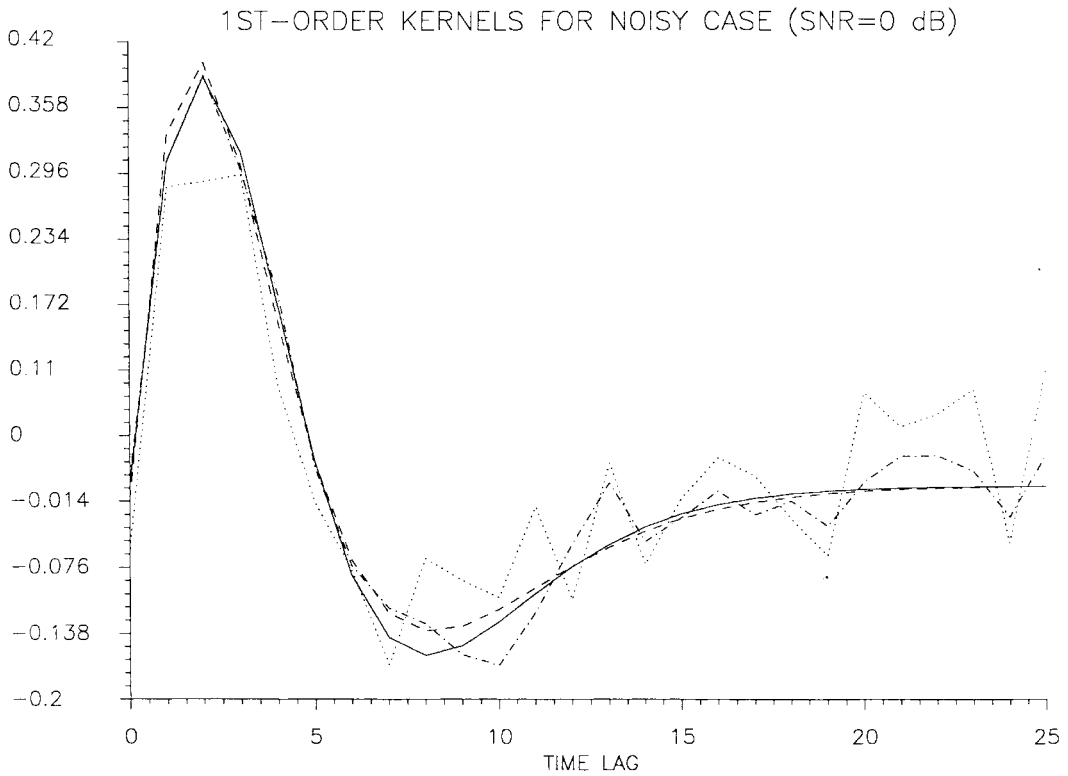


Fig. 7. The exact first-order kernel (solid line) and the three estimates obtained in the noisy case (SNR= 0 dB) via LET (dashed line), SVN (dot-dashed line), and TLP (dotted line). The LET estimate is the best in this example, followed closely by the SVN estimate in terms of accuracy—although requiring more computing time for training. The TLP estimate (obtained with four hidden units) is the worst in accuracy and computationally most demanding.

improves with increasing number K of hidden units, although the incremental improvement gradually diminishes. Since the computational burden for training increases with increasing K, we are faced with an important tradeoff: improvement in accuracy versus additional computational burden. By varying K, we determine a reasonable compromise for a TLP with four hidden units, where the number of free parameters is 112 and the required training time is about 20 times longer than SVN (or 400 times longer than LET). The resulting TLP kernel estimates are not as accurate as the SVN or LET estimates, as illustrated in Figs. 5 and 6 for the first-order and the second-order kernels, respectively. Note that SVN training required 200 backprop iterations in this example versus 2000 iterations required for TLP training. Thus, SVN appears superior to TLP in terms of accuracy and computational effort in this example of a second-order Volterra system.

To make the comparison more relevant to actual applications, we add independent Gaussian white noise to the output data for a signal-to-noise ratio of 0 dB (i.e., the noise variance is equal to the noise-free output mean-square value). The obtained first-order kernel estimates via the three methods (LET, SVN, TLP) are shown in Fig. 7 along with the exact kernel, and the obtained second-order kernel estimates are shown in Fig. 8. The LET estimates are the most accurate and quickest to obtain, followed closely by the SVN estimates in terms of accuracy—although SVN requires longer computing time (by a factor of 20). The TLP estimates are clearly inferior to either LET or SVN estimates in this example and require longer computing time (about 20 times longer for $K = 4$). The kernel estimates are used here as the means of comparison. These results demonstrate the considerable benefits of using SVN configurations instead of TLP for Volterra system modeling purposes, although there may be some cases where the TLP configuration has a natural advantage, e.g., systems with sigmoidal output nonlinearities.

In the chosen example, LET appears to yield the best model and associated kernel estimates. However, its application is practically limited to low-order kernels (up to third) and, therefore, it is the preferred method only for systems with low-order nonlinearities. On the other hand, SVN offers not only an attractive alternative for low-order kernel estimation and modeling, but also a *unique practical solution when the system nonlinearities are of high order*. The latter constitutes the primary motivation for proposing the SVN configuration for nonlinear system modeling.

To demonstrate this important point, we consider an arbitrarily defined high-order nonlinear system described by the output equation

$$y = \left(v_1 + 0.8v_2^2 - 0.6v_1^2 v_2\right) \sin[(v_1 + v_2)/5] \tag{26}$$

where the "internal"variables (v_1, v_2) are given by the difference equations

$$v_1(n) = 1.2v_1(n-1) - 0.6v_1(n-2) + 0.5x(n-1) \tag{27}$$

$$v_2(n) = 1.8v_2(n-1) - 1.1v_2(n-2) + 0.2v_2(n-3) + 0.1x(n-1) + 0.1x(n-2) \tag{28}$$

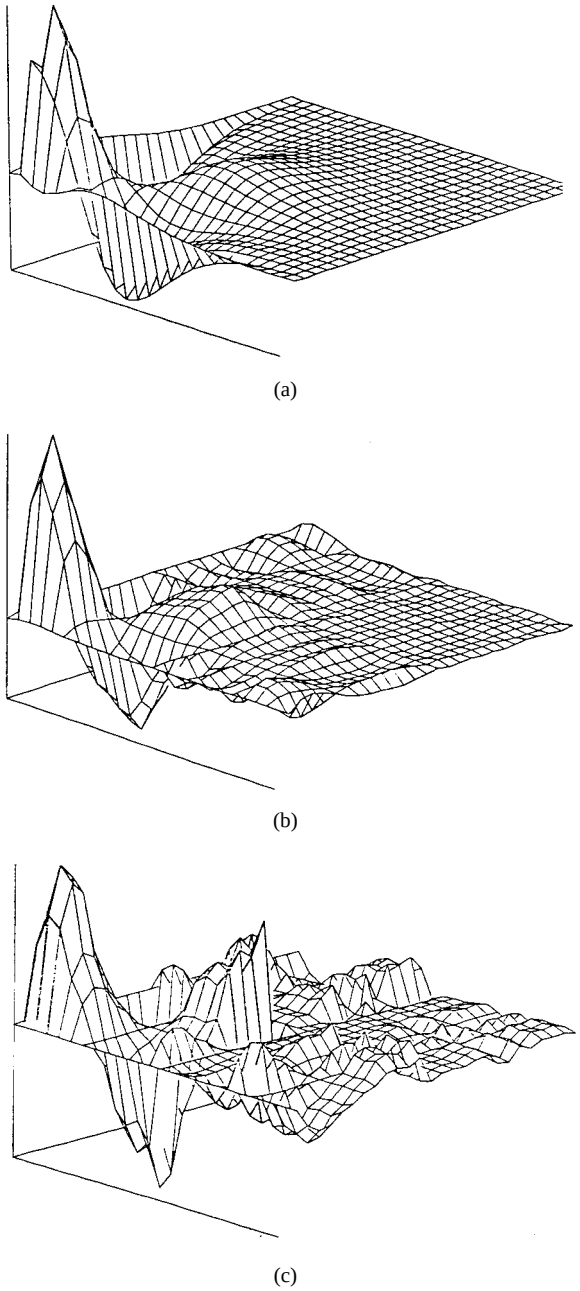


Fig. 8. The second-order kernel estimates obtained in the noisy case (SNR= 0 dB) via LET (a), SVN (b), TLP (c). Relative performance is the same as described in the case of first-order kernels (see caption of Fig. 7).

and $x(n)$ denotes the discrete-time input signal that is chosen in this simulation to be a 1024-point segment of Gaussian white noise with unit variance. Use of LET with six Laguerre functions to estimate truncated second- and third-order Volterra models yields output predictions with normalized mean-square errors (NMSE) of 47.2% and 34.7%, respectively. Note that the obtained kernel estimates are seriously biased because of the presence of higher order terms in the output equation that are treated by LET as correlated residuals in least-squares estimation. Use of the SVN approach (employing five hidden units of seventh-degree) yields a model of improved prediction accuracy (NMSE $= 6.1\%$) and mitigates the problem of kernel estimation bias by allowing estimation of nonlinear terms up to seventh-order (note that the selected system is of infinite order, i.e., it has Volterra kernels of all orders, although of gradually diminishing size). Training of this SVN with the aforementioned data required 5000 iterations and much longer computing time than LET (about 300 times longer). Training of a TLP model with these data yielded less prediction accuracy than the SVN for comparable numbers of hidden units and free parameters. For instance, a TLP with eight hidden units yields an output NMSE of 10.3% (an error that can be gradually but slowly reduced by increasing the number of hidden units). Note that the number of free parameters in this example is: 225 and 165 for TLP and SVN, respectively [see (21) and (23)]. An illustration is given in Fig. 9, where the model predictions for the five-unit SVN, the eight-unit TLP and the third-order LET are shown along with the actual system output for a segment of test data.

The important practical issues of how we determine the appropriate number of hidden units and the appropriate degree of polynomial nonlinearity in the activation functions are addressed by preliminary experiments and successive trials. For instance, the degree of polynomial nonlinearity can be established by preliminary testing of the system under study with sinusoidal inputs and subsequent determination of the highest harmonic in the output via fast Fourier transform (FFT), or by varying the power level of a white-noise input and fitting the resulting output variance to a polynomial expression [21], [22]. On the other hand, the number of hidden units can be determined in general by successive trials in ascending order (i.e., adding new units and observing the amount of error reduction) or by pruning methods that eliminate units with weak polynomial coefficients. Note that the input weights in the SVN are kept normalized to unity Euclidean norm for each hidden unit; thus the polynomial coefficients are a true measure of the relative importance of each hidden unit. A systematic method for determining the required number of hidden units for second-order Volterra systems has been proposed via eigen-decomposition (principal dynamic modes) in [27].

VI. Conclusion

Volterra models of nonlinear systems constitute a canonical representation for a broad class of systems and offer a general mathematical framework to assess the relative efficacy of various network implementations for nonlinear mapping of input vectors onto output scalars (or vectors). A popular class of feedforward neural networks, the TLP shown in Fig. 1, was analyzed in the Volterra context and compared to a network architecture, termed SVN, that employs polynomial activation functions, as shown in Fig. 3. The latter is shown to result from the MVM obtained from discrete kernel expansions (see Fig. 2). Comparisons were made with respect to the relative efficacy of these representations, and conditions for their functional equivalence were explored. It was shown that the general discrete-time Volterra model is functionally equivalent to a TLP when the number of its hidden units tends to infinity.

Although all three approaches (TLP, SVN, MVM) may represent the input–output relation of nonlinear systems, their

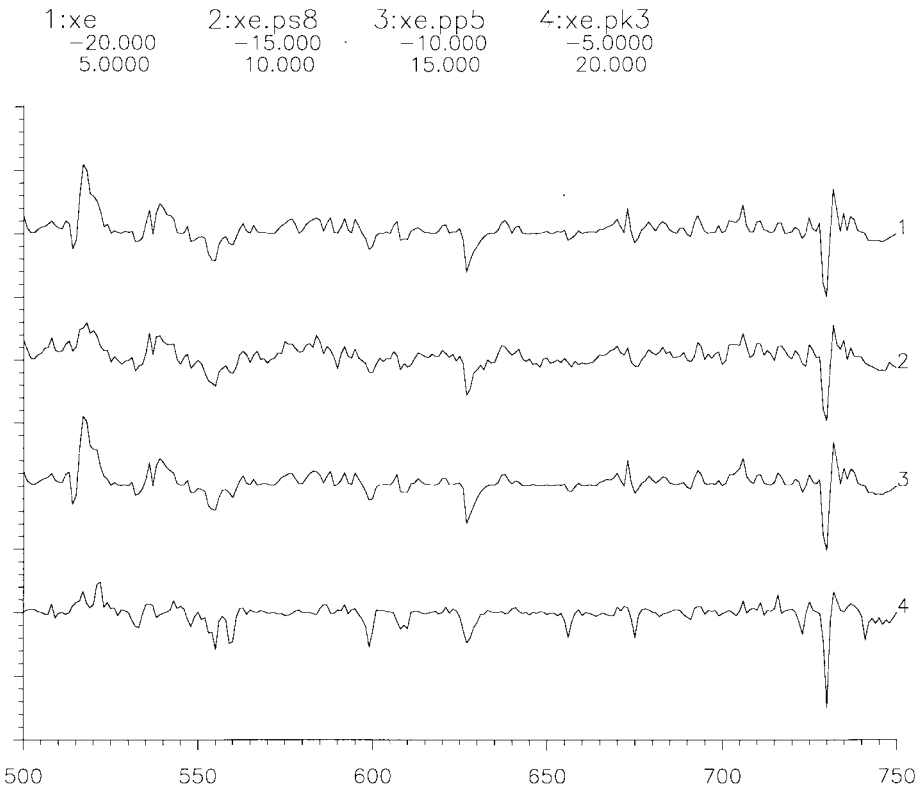


Fig. 9. The model predictions for a high-order system using LET with a truncated third-order Volterra model (trace 4), SVN with five hidden units of seventh-degree (trace 3), and TLP with eight hidden units (trace 2), along with the exact system output (trace 1).

relative efficiency (i.e., number of free parameters and the required computational effort for estimation/training) may vary dramatically from application to application. For low-order Volterra systems, the traditional approach of kernel estimation (using recent techniques, such as LET) appears to be most efficient. For high-order Volterra systems, where kernel estimation is impractical, SVN is more efficient than TLP for nonsigmoidal output nonlinearities or whenever curvilinear "decision boundaries" are involved.

The development of efficient SVN models benefits from judicious selection of the number of hidden units, which can be assisted by preliminary estimation of the system "principal dynamic modes" using a truncated quadratic Volterra model [27]. The advantages of SVN were demonstrated in avoiding significant bias in estimating kernels of truncated models for high-order systems and in rendering feasible the daunting challenge of high-order nonlinear system modeling. This paper aims at instigating interest in the study of the relative strengths and weaknesses of the two approaches, with the goal of their cooperative use for mutual benefit.

REFERENCES

[1] E. K. Blum and L. K. Li, "Approximation theory and feedforward networks," *Neural Networks*, vol. 4, pp. 511–515, 1991.

[2] T. Chen and H. Chen, "Universal aaproximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems," *IEEE Trans. Neural Networks*, vol. 6, pp. 911–916, 1995.

[3] S. Chen, G. J. Gibson, and C. F. N. Cowan, "Adaptive channel equalization using a polynomial perceptron structure," in *Proc. Inst. Elec. Eng.*, 1990, vol. 137, pp. 257–264.

[4] M. S. Chen and M. T. Manry, "Conventional modeling of the multilayer perceptron using polynomial basis functions," *Neural Networks*, vol. 4, pp. 164–166, 1993.

[5] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Contr., Signals, Syst.*, vol. 2, pp. 303–314, 1989.

[6] G. W. David and M. L. Gasperi, "ANN modeling of Volterra systems," in *Proc. IJCNN'91*, Seattle, WA, 1991, pp. II 727–734.

[7] R. J. P. de Figueiredo, "Implications and applications of Kolmogorov's superposition theorem," *IEEE Trans. Automat. Contr.*, vol. 25, pp. 1227–1231, 1980.

[8] ——, "A generalized Fock space framework for nonlinear system and signal analysis," *IEEE Trans. Circuits Syst.*, vol. CAS-30, pp. 637–647, Sept. 1983 (Special invited issue on nonlinear circuits and sytems).

[9] ——, "A new nonlinear functional analytic framework for modeling artificial neural networks," (invited paper), in *Proc. 1990 IEEE Int. Symp. Circuits Syst.*, New Orleans, LA, May 1990, pp. 723–726.

[10] ——, "An optimal mutlilayer neural interpolating (OMNI) net in a generalized Fock space setting," (invited paper), in *Proc. IJCNN*, Baltimore, MD, June 1992.

[11] K.-I. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Networks*, vol. 2, pp. 183–192, 1989.

[12] G. Govind and P. A. Ramamoorthy, "Multilayered neural networks and Volterra series: The missing link," in *Proc. Int. Conf. Syst. Eng.*, 1990, pp. 633–636.

[13] M. H. Hassoun, *Fundamentals of Artificial Neural Networks*. Cambridge, MA: MIT Press, 1995.

[14] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.

[15] W. G. Knecht, "Nonlinear noise filtering and beamforming using the perceptron and its Volterra approximation," *IEEE Trans. Speech Audio Processing.*, vol. 2, pp. 55–62, 1994.

[16] A. N. Kolmogorov, "On the representation of continuous functions of several variables by superposition of continuous functions of one variable and addition," *Dokl. Akad. Nauk*, SSSR, vol. 114, pp. 953–956, 1957; *AMS Transl.*, vol. 2, pp. 55–59, 1963.

[17] M. J. Korenberg, "Parallel cascade identification and kernel estimation for nonlinear systems," *Ann. Biomed. Eng.*, vol. 19, pp. 429–455, 1991.

[18] S. Y. Kung, *Digital Neural Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[19] M. Leshno, V. Lin, A. Pinkus, and H. White, "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function," *Neural Networks*, vol. 6, pp. 861–867, 1993.
[20] G. G. Lorentz, "The 13th problem of Hilbert," in *Proc. Symp. Pure Math.*, F. E. Browder, Ed., vol. 28. Providence, RI: Amer. Math. Soc., 1976, pp. 419–430.
[21] P. Z. Marmarelis and V. Z. Marmarelis, *Analysis of Physiological Systems: The White Noise Approach*. New York: Plenum, 1978. Russian translation: Mir Press, Moscow, 1981. Chinese translation: Academy of Sciences Press, Beijing, 1990.
[22] V. Z. Marmarelis, Ed., *Advanced Methods of Physiological System Modeling*, vol. I. Los Angeles: Univ. Southern California, Biomedical Simulations Resource, 1978.
[23] ———, *Advanced Methods of Physiological System Modeling*, vol. II. New York: Plenum, 1989.
[24] ———, *Advanced Methods of Physiological System Modeling*, vol. III. New York: Plenum, 1994.
[25] V. Z. Marmarelis, "Signal transformation and coding in neural systems," *IEEE Trans. Biomed. Eng.*, vol. 36, pp. 15–24, 1989.
[26] ———, "Identification of nonlinear biological systems using Laguerre expansions of kernels," *Ann. Biomed. Eng.*, vol. 21, pp. 573–589, 1993.
[27] ———, "Nonlinear modeling of physiological systems using principal dynamic modes," in *Advanced Methods of Physiological System Modeling*, vol. III. New York: Plenum, 1994, pp. 1–28.
[28] V. Z. Marmarelis and M. E. Orme, "Modeling of neural systems by use of neuronal modes," *IEEE Trans. Biomed. Eng.*, vol. 40, pp. 1149–1158, 1993.
[29] V. Z. Marmarelis, M. C. Citron, and C. P. Vivo, "Minimum-order Wiener modeling of spike-output systems," *Biol. Cybern.*, vol. 54, pp. 115–123, 1986.
[30] G. Palm, "On representation and approximation of nonlinear systems. Part II: Discrete systems," *Biol. Cybern.*, vol. 34, pp. 49–52, 1979.
[31] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Washington, D.C.: Spartan Books, 1962.
[32] D. E. Rumelhart and J. L. McClelland, Eds., *Parallel Distributed Processing*, vols. I and II. Cambridge, MA: MIT Press, 1986.
[33] I. W. Sandberg, "Approximation theorems for discrete-time systems," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 564–566, 1991.
[34] S. K. Sin and R. J. P. de Figueiredo, "Efficient learning procedures for optimal interpolative nets," *IEEE Trans. Neural Networks*, vol. 6, pp. 90–113, 1993.
[35] D. F. Specht, "Probabilistic neural networks and the polynomial adaline as complementary techniques for classification," *IEEE Trans. Neural Networks*, vol. 1, pp. 111–121, 1990.
[36] D. A. Sprecher, "An improvement in the superposition theorem of Kolmogorov," *J. Math. Anal. Applicat.*, vol. 38, pp. 208–213, 1972.
[37] N. Wiener, *Nonlinear Problems in Random Theory*. New York: The Technology Press of M.I.T. and Wiley, 1958.
[38] J. Wray and G. G. R. Green, "Calculation of the Volterra kernels of nonlinear dynamic systems using an artificial neural network," *Biol. Cybern.*, vol. 71, pp. 187–195, 1994.
[39] Z. Xiang, G. Bi, and T. Le-Ngoc, "Polynomial perceptrons and their applications to fading channel equalization and cochannel interference suppression," *IEEE Trans. Signal Processing*, vol. 42, pp. 2470–2479, 1994.

Vasilis Z. Marmarelis (M'79–SM'94–F'97) was born in Mytilini, Greece, on November 16, 1949. He received the Diploma degree in electrical and mechanical engineering from the National Technical University of Athens in 1972 and the M.S. and Ph.D. degrees in engineering science (information science and bioinformation systems) from the California Institute of Technology, Pasadena, in 1973 and 1976, respectively.

After two years of postdoctoral work at the California Institute of Technology, he joined the faculty of Biomedical and Electrical Engineering at the University of Southern California, Los Angeles, where he is currently Professor and Director of the Biomedical Simulations Resource, a research center funded by the National Institutes of Health since 1985 and dedicated to modeling/simulation studies of biomedical systems. He served as Chairman of the Biomedical Engineering Department from 1990 to 1996. His main research interests are in the areas of nonlinear and nonstationary system identification and modeling, with applications to biology, medicine, and engineering systems. Other interests include spatiotemporal and nonlinear/nonstationary signal processing, and analysis of neural systems and networks with regard to information processing. He is coauthor of the book *Analysis of Physiological Systems: The White-Noise Approach* (New York: Plenum, 1978; Russian translation: Moscow, Mir Press, 1981; Chinese translation: Academy of Sciences Press, Beijing, 1990) and editor of three volumes on *Advanced Methods of Physiological System Modeling* (1987, 1989, 1994). He has published more than 100 papers and book chapters in the area of system and signal analysis.

Xiao Zhao (S'92–M'95) received the B.S. degree in physics from Fudan University, Shanghai, China, in 1982, the M.S. degree in biomedical engineering from the University of Sciences and Technology of China, Beijing, China, in 1985, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, in 1994.

From 1994 to 1995 he was a Staff Scientist in the advanced signal and image processing group at Physical Optics Corporation in Torrance, CA. Since 1995, he has been with the Biosciences and Bioengineering Division of the Southwest Research Institute in San Antonio, TX. His primary research interests include signal processing, pattern recognition, and nonlinear system modeling.