Annals of Biomedical Engineering, Vol. 24, pp. 250–268, 1996
Printed in the USA. All rights reserved.

0090-6964/96 $10.50 + .00

The Identification of Nonlinear Biological Systems: Volterra Kernel Approaches

MICHAEL J. KORENBERG* and IAN W. HUNTER†

*Department of Electrical and Computer Engineering, Queen's University, Kingston, Ontario, Canada, and †Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA

Abstract—Representation, identification, and modeling are investigated for nonlinear biomedical systems. We begin by considering the conditions under which a nonlinear system can be represented or accurately approximated by a Volterra series (or functional expansion). Next, we examine system identification through estimating the kernels in a Volterra functional expansion approximation for the system. A recent kernel estimation technique that has proved to be effective in a number of biomedical applications is investigated as to running time and demonstrated on both clean and noisy data records, then it is used to illustrate identification of cascades of alternating dynamic linear and static nonlinear systems, both single-input single-output and multivariable cascades. During the presentation, we critically examine some interesting biological applications of kernel estimation techniques.

Keywords—System identification, Volterra kernels, Functional expansions, Nonlinear systems, Volterra series, Cascade Analysis.

INTRODUCTION

As early as 1887, Volterra was considering the general problem of functions that operate on functions. This work, his subsequent elaborations (94,95), and the developments of others (in particular, Frechet, 17) have grown into a field of mathematics whose primary interest is the study of functionals and operators and the representation of nonlinear systems. In such work, Volterra kernels (a constant and one-dimensional and multidimensional weighting functions) play a central part in representing the nonlinear system by a Volterra series.

Acknowledgment—We gratefully acknowledge the computer programming assistance provided by Tilemachos Doukoglou and Serge Lafontaine. This research was supported by grants from the Medical Research Council of Canada, the Natural Sciences and Engineering Research Council of Canada, and the Institute for Robotics and Intelligent Systems (a Canadian Network of Centers of Excellence).

Address correspondence to M. J. Korenberg, Department of Electrical and Computer Engineering, Queen's University, Kingston, Ontario, Canada, or to I. W. Hunter, Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Room 3-154, Cambridge, MA 02139, U.S.A.

(Received 29Mar94, Revised 26Jun95, Accepted 5Oct95)

Volterra Functional Expansions

In a previous paper (47), we concentrated on the use of Wiener's functional expansion to characterize a nonlinear system, discussing the advantages, limitations, and biological applications. In this paper, we study in brief Volterra series (or functional expansions) and their use in representing or approximating nonlinear biological systems. We also examine an accurate method for estimating the kernels (showing running times and performance on noisy data) and its application to identifying cascades of alternating dynamic linear and static nonlinear elements for both single-input single-output (SISO) and multivariable cascades. Once a functional expansion or other model has been identified, its validity (6) and reliability should be investigated. We, therefore, point out some pitfalls that can unsettle even the best-intentioned attempt to estimate accurately the kernels characterizing a system and illustrate our comments with critical references to the literature.

A functional, F, transforms values of the input defined over the input domain (*e.g.*, time) into a value of the output defined at a single point of the output domain (*e.g.*, at a fixed instant). An operator, S, transforms values of the input defined over the input domain (*e.g.*, time) into values of the output defined over the output domain (*e.g.*, time). To quote Schetzen (80), "the essential difference is that in the functional viewpoint we focus our attention on the output at a particular instant in time, while in the operator viewpoint we focus our attention on the complete output function." It is important to note that for static (*i.e.*, zero-memory or nondynamic) systems the output, y, is said to be a function of the input, x, whereas, for dynamic systems (*i.e.*, systems with memory), y is said to be a functional of x. Many nonlinear differential or integro-differential equations have Volterra series solutions that hold for suitable restrictions on the input signal, for example, imposing a bound on the input amplitude. Barrett (4) has shown how to find both the Volterra series solution and the region of its convergence for certain classes of nonlinear differential equations. He shows that the Volterra series solution often can be generated by regarding the

nonlinear terms in the differential equation as a "perturbation" on the linear system defined by the linear terms in the equation, then obtaining the convolution integral (see below) solution, which will contain the nonlinear perturbation terms, then repetitively substituting for those nonlinear terms in the convolution integral that involve the output.

Consider a functional in which the value of y at some fixed instant, for example $t = t_1$ depends on values of x over some finite interval, for example, $t_1 - T \leq t \leq t_1$, that is,

$$y(t_1) = F[x(t),\ t_1 - T \leq t \leq t_1]. \tag{1}$$

Volterra considered analytic functionals, in which $y(t)$ (*e.g.*, at $t = t_1$) could be represented exactly as

$$y(t) = \sum_{i=0}^{Q} V_i[h_i,x] \tag{2}$$

where Q can be infinite, and the zero-order Volterra functional is the constant

$$V_0[h_0,x] = h_0 \tag{3}$$

and for $i \geq 1$, the i-th-order Volterra functional is

$$V_i[h_i,x] = \int_{-\infty}^{\infty} \cdots \int h_i(\tau_1, \ldots, \tau_i)\, x(t - \tau_1) \ldots x(t - \tau_i)\, d\tau_1 \ldots d\tau_i \tag{4}$$

The functional series (Eq. 2) has become known as the Volterra series or Volterra functional expansion (see reviews in 56,59,66,79).

Volterra Kernels

The functions $h_0, h_1, \ldots$ in Eq. 2 are called Volterra kernels, and, in particular, h_i is called the i-th order Volterra kernel.

It is useful to note the following points:

1. The zeroth-order Volterra kernel is a constant equal to the zero-input response of the system. Note that the input may represent a perturbation about some mean level. Then the zero-input response would be the response to this mean level, assumed applied for all $t > -\infty$. For an interpretation of the zeroth-order kernel using standard perturbation theory, see Reference 60.
2. If the system is causal (sometimes called physically realizable or nonanticipatory), then the kernel $h_i(\tau_1, \ldots, \tau_i) = 0$ when any of the $\tau_1, \ldots, \tau_i$ is less than zero, and the lower limits of the integrals in Eq. 4 can be set equal to zero. If, in addition, the system has finite memory T, then the upper limits are set equal to T (because the output can depend on input delays up to, at most, T). In Volterra's original formulation, the integrals in Eq. 4 generally were taken between two finite limits.
3. With no loss in generality, it can be assumed (95) that each kernel $h_i(\tau_1, \ldots, \tau_i)$ in Eq. 4 is symmetric with respect to any permutation of $\tau_1, \ldots, \tau_i$. Two nonsymmetric forms of the Volterra kernel, namely triangular and regular, sometimes are used (66). For example, when the triangular form is used the integrations in Eq. 4 with respect to $\tau_1, \tau_2, \ldots, \tau_{i-1}, \tau_i$ frequently are taken from $-\infty$ (or 0 for a causal system) to, respectively, $\tau_2, \tau_3, \ldots, \tau_i, T$, where T is the memory of the system. The regular form is closely related to the triangular kernel and essentially is obtained by replacing $\tau_1, \tau_2, \ldots, \tau_{i-1}, \tau_i$ with, respectively, $\lambda_1 + \ldots + \lambda_i, \lambda_2 + \ldots + \lambda_i, \ldots, \lambda_{i-1} + \lambda_i, \lambda_i$ (for additional details, see 66).

The functional $V_i[h_i,x]$ is called an i-th degree homogeneous Volterra functional, because if x is replaced by cx (where c is a constant) then $V_i[h_i,cx] = c^i V_i[h_i,x]$.

Frechet (17) showed that if at, for example, $t = t_1$,

1. $y(t)$ is a functional (not necessarily analytic) of x and depends on values of $x(t)$ only over a finite interval, for example, $t_1 - T \leq t \leq t_1$, as in Eq. 1; and
2. F is a continuous functional of the input, so that small changes in the input result in small changes in $y(t_1)$; and
3. the portion of x defined over the interval $t_1 - T \leq t \leq t_1$ belongs to a compact set of signals (defined over this interval); that is, the set of signals is uniformly bound and equicontinuous over this time interval (30);

then, at $t = t_1$, $y(t)$ can be approximated uniformly for any $x(t)$, $t_1 - T \leq t \leq t_1$, belonging to the compact set, by the right side of Eq. 2, to any given accuracy for sufficiently large but finite Q (see also 76).

Note that the integrations in Eq. 4 are conducted over the interval $[0,T]$.

The Use of Volterra Series for System Representation

We now shift our attention from the value of y at one fixed instant t_1 (a functional viewpoint) to the values attained by y for all $t_1 \geq 0$ (an operator viewpoint).

Suppose (after Barrett (5))

1. the above assumptions hold for all $t_1 \geq 0$; and
2. the system is time-invariant (*i.e.*, a translation of the input over time results in an equivalent translation of the output over time); and
3. the input x belongs to a set of signals that is uniformly bounded and equicontinuous for all $t \geq -T$;

then $y(t)$ for each $t \geq 0$ can be approximated uniformly over this set of input signals to any given degree of accuracy by the right side of Eq. 2 for sufficiently large but finite Q (which does not depend on the particular input x selected from the set or on the value of $t \geq 0$).

Therefore, in summary, to any given accuracy.

$$y(t) = h_0 + \int_0^T h_1(\tau_1)\, x(t - \tau_1)\, d\tau_1 + \int_0^T\int h_2(\tau_1,\tau_2)\, x(t - \tau_1)x(t - \tau_2)\, d\tau_1 d\tau_2 + \ldots + \int_0^T\cdots\int h_Q(\tau_1, \ldots, \tau_Q) x(t - \tau_1) \ldots x(t - \tau_Q)\, d\tau_1 \ldots d\tau_Q + \varepsilon(t) \quad (5)$$

where, for any x from the input set and any $t \geq 0$, the error magnitude $|\epsilon(t)|$ can be made arbitrarily small for sufficiently large Q.

This Volterra series approximation can be used to represent, over a uniformly bounded equicontinuous set of inputs, any nonlinear system that is finite-memory, physically realizable, time-invariant, single-input single-output, and a continuous functional of its input. Moreover, Eq. 5 may be generalized to the multiple-input multiple-output and time-varying cases (56). However, in general, infinite-memory systems, such as those incorporating hysteresis or limit-cycle properties, cannot be represented by the Volterra series and are outside the scope of this paper. The identification of systems exhibiting hysteresis has been considered by several authors, for example, Andronikou *et al.* (2).

In place of the finite-memory assumption, Boyd and Chua (8) have defined systems with "fading memory," which they show also can be represented by Volterra series. Note that the case $T = \infty$ has been used to include inputs that are not zero for distant negative time and systems that depend, to an arbitrarily small extent, on the remote past of the input. As an example of the latter, consider the following analytic system, described by

$$y(t) = \int_0^\infty e^{-\tau} x(t - \tau)\, d\tau + \int_0^\infty\int e^{-\tau_1-\tau_2} x(t - \tau_1)\, x(t - \tau_2)\, d\tau_1 d\tau_2$$

In this example the first- and second-order kernels are decaying exponentials, so that input values in the remote past have a negligible effect on the output when the input again is a member of the uniformly bounded equicontinuous set of signals.

It is important to note that linear time-invariant systems form a subclass of systems describable by the right side of Eq. 5, because the convolution integral defines the output as

$$y(t) = \int_0^\infty h(\tau)\, x(t - \tau)\, d\tau \quad (6)$$

The first-order kernel $h(\tau)$ in Eq. 6 is known as the unit impulse response function (or, equivalently, the system weighting function, inverse Laplace transform of the transfer function, or inverse Fourier transform of the frequency response function). This kernel is a one-dimensional weighting function that determines the extent to which the input value τ seconds earlier contributes to the present value of the output. More generally, the i-th-order kernel $h_i(\tau_1, \ldots, \tau_i)$ is an i-dimensional weighting function that determines the extent to which the interaction of input values that are $\tau_1, \ldots, \tau_i$ seconds earlier contributes to the present output value.

As discussed, an analytic nonlinear system can be described exactly by the right side of Eq. 5, when $\epsilon(t) \equiv 0$ and Q may be infinite. Note that, for an analytic nonlinear system, the first-order Volterra kernel, $h_1(\tau)$, in general is not equal to the unit impulse response. Rather, the output caused by a unit impulse is given by

$$y(t) = h_0 + h_1(t) + h_2(t,t) + \ldots + h_Q(t, \ldots, t). \quad (7)$$

The influence of nonlinearity on this response is evident through the additional terms h_0, $h_2(t,t)$, . . . , $h_Q(t, \ldots, t)$.

To model a nonlinear system by using a truncated Volterra series (more particularly, using the right side of Eq. 5 when Q is finite) the kernels h_0, $h_1(\tau)$, . . . , $h_Q(\tau_1, \ldots, \tau_Q)$ must be determined.

METHODS FOR DETERMINING THE VOLTERRA KERNELS

Various techniques have been used to analyze the Volterra series or to estimate the Volterra kernels (*e.g.*, see 1,7,9,10,15,16,19,22,38,41,43,45). Schetzen (78,79) proposed a Volterra kernel estimation method that involved the use of multiple pulse inputs and repeated experiments to identify the Volterra kernels of a finite-order system. This valuable approach has been applied by Stark (85) as one method for estimating the kernels of the pupillary system. In addition, Pece *et al.* (63) used Schetzen's method to estimate the first- and second-order Volterra kernels for a dark-adapted locust photoreceptor intracellular membrane potential response. These kernels were similar, except for a change in time scale, to the Wiener kernels estimated (via the fast orthogonal algorithm; 38) from white noise experiments at various background light intensities.

We illustrate Schetzen's (78) method in brief by estimating the Volterra kernels of a second-order (analytic) nonlinear system by using double-pulse experiments. The pulses are sufficiently brief to approximate impulses, and,

for convenience, are assumed to have unit area. Suppose that a single such pulse is applied to the (resting) system at $t = 0$. Then, the resultant output is

$$y_1(t) = h_0 + h_1(t) + h_2(t,t). \tag{8}$$

Next, suppose that the resting system is disturbed by a pulse applied at $t = 0$ followed by a pulse at $t = T$, where $T > 0$. The resultant output is

$$\begin{aligned} y_2(t) &= h_0 + h_1(t) + h_2(t,t) + h_1(t - T) \\ &\quad + h_2(t - T,t - T) + 2h_2(t,t - T) \\ &= y_1(t) + y_1(t - T) - h_0 + 2h_2(t,t - T). \end{aligned} \tag{9}$$

The zero-order kernel h_0 is simply the zero-input response and is assumed to be known. Therefore, we can solve Eq. 9 for $h_2(t,t - T)$, $T > 0$, and, therefore, obtain the second-order kernel, except along its main diagonal. The diagonal kernel values can be obtained by interpolation, assuming continuity of the kernel. Then, Eq. 8 can be solved for $h_1(t)$, the first-order kernel.

Note that kernel continuity is not guaranteed for every nonlinear system that is a continuous functional of its input. Thus a system can be a continuous functional, in that "small" changes in the system input result in "small" changes in the system output, yet the system's kernels need not be continuous functions. An example of this is a Hammerstein model, which has non-zero kernel values along only the main diagonals of its Volterra kernels. It is clear that, in this case, the diagonal values of the second-order Volterra kernel cannot be interpolated from the zero off-diagonal values. A more elaborate example can be constructed by imagining that the system under consideration is the sum of a subsystem whose kernels are continuous and a subsystem that is a Hammerstein model. Again, the diagonal kernel values of the overall system could not be inferred from the off-diagonal values. One simple and widely known method for estimating diagonal kernel values (without assuming kernel continuity) is illustrated on our second-order nonlinear system.

One disadvantage of Schetzen's method is that the highest-order kernel must be estimated before the next highest order, *etc.* Schetzen (79) points out that a certain simplification is possible by suppressing the response of all of the even-order, or all of the odd-order, kernels by applying the inputs x and $-x$ separately. Therefore, to determine just the first-order kernel of a fifth-order system first requires determining all higher-order kernels (or all odd higher-order kernels, if the simplification is used) in descending sequence starting with the fifth-order kernel.

When kernel continuity cannot be assumed, one can estimate $h_1(t)$ and $h_2(t,t)$ of our second-order nonlinear system by applying, for example, an additional single pulse having an area of 2 at $t = 0$ to the system at rest. The resultant output would be

$$y_3(t) = h_0 + 2h_1(t) + 4h_2(t,t). \tag{10}$$

Because, as noted above, h_0 is known from the zero-input response, $h_1(t)$ and $h_2(t,t)$ can be obtained by simultaneously solving Eqs. 8 and 10.

Korenberg (31) devised a method, based on Volterra's original definition of the kernels, to estimate a kernel of any given order without requiring the previous estimation of other kernels. Variations of the method (35) involving exponentially or sinusoidally modulated random inputs were devised to cope with data corrupted by output noise. These methods work best for identifying lower-order kernels (although the system order may be arbitrarily high). In addition, the Volterra kernels can be calculated very accurately by using the parallel cascade method (34,37,41,43), which is effective even for higher-order nonlinear systems and systems with lengthy memory. The sinusoidally modulated random inputs used (35) for kernel estimation are examples of cyclostationary inputs, which subsequently, also were considered for the same purpose by Gardner and Archer (18).

ADVANTAGES AND DISADVANTAGES OF THE VOLTERRA APPROACH

1. A prime advantage of the functional expansion approximation is its generality.
2. Any finite-memory system that is a continuous functional of its input can be approximated pointwise to an arbitrary degree of accuracy by a Volterra series of sufficient but finite order Q for a given set of input signals, x, which are uniformly bounded and equicontinuous.
3. One difficulty is that the order, Q, may need to be very large to achieve a specified accuracy over the given set of signals (5). The determination of even the first- and second-order Volterra kernels can involve the estimation of a large parameter set. For example, Marmarelis and Naka (55) computed over 250 distinct second-order kernel values (actually a very moderate number for this type of analysis but more than the number of parameters that, say, a differential equation model would typically require) in their elegant nonlinear system modeling of the receptive field responses in the catfish retina.
4. Directly identifying the Volterra kernels in Eq. 5 for general nonlinear systems poses severe numerical problems, in particular because the homogeneous functionals $V_i[h_i,x]$ in the series are not mutually orthogonal. Orthogonality would imply that

$$\begin{aligned} E\{V_i[h_i,x] \cdot V_j[h_j,x]\} &= 0 \text{ if } i \neq \text{j} \\ &> 0 \text{ if } i = \text{j}, \end{aligned}$$

where $E\{\cdot\}$ denotes infinite-time average.

ESTIMATION OF KERNELS IN A FUNCTIONAL SERIES APPROXIMATION USING WIENER'S ORTHOGONAL APPROACH

In 1955, Barrett (reviewed in 3) considered the general problem of generating multidimensional orthogonal polynomials for a wide variety of inputs (lacking special autocorrelation properties), before proceeding to the particular case of Gaussian processes for which the multidimensional Hermite polynomials formed an orthogonal set. In addition, Schetzen (79) considered the possibility of constructing orthogonal polynomials corresponding to specific non-Gaussian inputs for the purpose of nonlinear system identification. Palm and Poggio (62) note that the Gram-Schmidt orthogonalization of the Volterra series, performed by Wiener (97) for a Gaussian input process (more precisely, Brownian motion), can be conducted for many other processes. However, Palm and Poggio (62) also noted that calculating only a third-order orthogonal functional can produce severe difficulties, unless more specific assumptions on the input process are made. To understand the problem, consider Wiener's orthogonalization of the Volterra series for the Gaussian case.

Wiener Kernel Approach

In the late 1940s Norbert Wiener (reviewed in 50) realized the practical limitations of the nonorthogonal representation. In the most general form, the Volterra series is an infinite series, when Q in Eq. 5 is infinite. By using the Gram-Schmidt orthogonalization technique (see, for example, 87), Wiener orthogonalized the Volterra series, under the important assumption that the input is a Brownian process, which is the integral of a white (*i.e.*, has a flat power spectrum) Gaussian process. It is usual now to present Wiener's derivation in terms of a white Gaussian input. The resultant series, known as the Wiener series, can be written

$$y(t) = \sum_{j=0}^{\infty} G_j[w_j,x] \tag{11}$$

where the G_j are orthogonal functionals and the w_j are the Wiener kernels (discussed below). That is, if the input, x, is Gaussian white noise with a mean of zero and a given power spectral density, then

$$E\{G_k[w_k,x] \cdot G_j[w_j,x]\} = 0 \text{ if } k \neq j$$
$$> 0 \text{ if } k = j;$$

G_j is the j-th-order Wiener G-functional, where the G denotes that the functionals have been orthogonalized with respect to a particular Gaussian stationary, white input process.

Consider a discrete-time finite-memory system, which is a continuous mapping of its input (again, small changes in the input result in small changes in the output). Let the discrete Gaussian white input have mean zero and variance P. Via Gram-Schmidt orthogonalization (performed by Wiener for the continuous-time case), the first few discrete-time Wiener functionals can be shown to equal:

$$G_0[w_0,x] = w_0 \tag{12}$$

$$G_1[w_1,x] = \sum_{i=0}^{I} w_1(i)\, x(n-i) \tag{13}$$

$$G_2[w_2,x] = \sum_{i_1=0}^{I}\sum_{i_2=0}^{I} w_2(i_1,i_2)\, x(n-i_1)x(n-i_2) - P\sum_{i=0}^{I} w_2(i,i) \tag{14}$$

$$G_3[w_3,x] = \sum_{i_1=0}^{I}\sum_{i_2=0}^{I}\sum_{i_3=0}^{I} w_3(i_1,i_2,i_3)\, x(n-i_1)\, x(n-i_2)x(n-i_3) - 3P\sum_{i_1=0}^{I}\sum_{i_2=0}^{I} w_3(i_1,i_2,i_2)\, x(n-i_1) \tag{15}$$

where w_0, w_1, w_2, . . . are called the Wiener kernels, and w_j is the j-th-order Wiener kernel.

For the continuous-time case, Lee and Schetzen (51) showed that the Wiener kernels could be estimated from

$$w_j(\tau_1, \ldots, \tau_j) = \frac{1}{j!\,P^j} E\left\{\left(y(t) - \sum_{m=0}^{j-1} G_m[w_m,x]\right) x(t-\tau_1) \ldots x(t-\tau_j)\right\}.$$

When all the $\tau_1, \ldots, \tau_j$ are distinct (*i.e.*, unequal) this reduces to (51)

$$w_j(\tau_1, \ldots, \tau_j) = \frac{1}{j!\,P^j} E\{y(t)\, x(t-\tau_1) \ldots x(t-\tau_j)\}.$$

For the above-considered discrete-time case, t, τ_1, . . . , τ_j are replaced respectively by n, i_1, . . . , i_j, and the infinite-time average will be of $y(n)$ multiplied by the factors $x(n-i_1)$, . . . , $x(n-i_j)$, for distinct i_1, . . . , i_j. Barrett (5) and Goussard *et al.* (20) have pointed out that if these factors are replaced by the multidimensional Grad-Hermite polynomials of the factors, then $w_j(i_1,$

. . . , i_j) also can be determined at minor and major diagonal values. A similar idea for the continuous case is set forth by Lee and Schetzen (51). Goussard *et al.* (21) have shown that a stochastic approximation algorithm can be used to produce kernel estimates that are better than those from cross-correlation for the same computational complexity.

Watanabe and Stark (96) proposed a technique of kernel estimation in which each kernel was expanded by using a set of basis functions, such as the Laguerre functions; the coefficients in the resultant expansion were determined by least-squares fitting over the data record. When the kernels can be approximated accurately by using only a few Laguerre functions, the Watanabe-Stark method enables rapid estimation of kernels, and in their application, third-order kernels could be estimated in only a few seconds. Although the Laguerre function approach harks back to Wiener (97), Watanabe and Stark's (96) use of least-squares fitting introduced much greater accuracy. Note that least-squares estimates are obtained for the expansion coefficients and not, in general, for the resultant kernel estimates. When the number of Laguerre functions required to approximate the kernels accurately is considerably less than the system memory (see below), then and only then does the method save significant computing time. One means of indirectly obtaining least-squares estimates of the required coefficients is by combining (37,47) basis function expansion of the kernels with exact orthogonalization (discussed below). This permits a concise subset of basis functions to be rapidly selected to approximate the kernels accurately. Simulations were provided in Reference 47.

Following Wiener (97), Ogura (61) extensively considered the use of discrete Laguerre functions to expand the kernels to simplify their estimation. Ogura (61) developed a fast algorithm for kernel estimation involving efficient formulas (Eq. 131 in 61) for recursively calculating the outputs of the Laguerre function filters when driven by the experimental input. The Watanabe-Stark (96) method of determining the coefficients in the Laguerre expansion of the kernels by least-squares fitting over the data record recently was used by Marmarelis (57). In addition, Marmarelis (57) used Ogura's technique (61) for recursively calculating the outputs of the Laguerre filters (Eqs. 13 and 14 in 57 are equivalent to Eq. 131 in 61). The Watanabe-Stark (96) method of least-squares fitting Laguerre functions to estimate the kernels, together with Ogura's efficient formulas (61), can, depending on the application, yield impressive results even for third-order kernels. It is of interest that Marmarelis (57) added the detail of suggesting a criterion for choosing a parameter in the Laguerre functions controlling their rate of decay. Moreover, although least-squares fitting the Laguerre expansion (96) certainly is not new, Marmarelis's paper (57) does have the merit of repopularizing this useful technique from decades ago.

Sometimes, the use of Laguerre functions may result in estimating a more compact set of parameters (namely, the coefficients in the Laguerre expansion of the kernels) than the set of all distinct kernel values. Other times, numerical difficulties may make the use of Laguerre functions less parsimonious than using all distinct kernel values. For example, consider a simple Wiener model comprising a linear filter

$$k(j) = e^{j/2}, j = 0, \ldots, 15$$
$$= 0, \text{elsewhere}$$

followed by the second-degree polynomial $(\cdot) + (\cdot)^2$. That is, if $u(n)$ is the output of the linear filter, then the output of the polynomial is $y(n) = u(n) + u^2(n)$.

Theoretically, because the memory length is 16, only 16 distinct Laguerre functions should be required to exactly represent the kernels by using a Laguerre expansion. But this may not result in accurate kernel estimation even with the use of noise-free data, which is investigated readily by using the comprehensive LYSIS software program (53), and the criterion in Reference 57 to choose the value of the parameter controlling the rate of decay of the Laguerre functions. However, using the fast orthogonal algorithm (38,43) to estimate kernels up to second-order with lags 0, . . . , 15 yields very good results. In this case, use of the Laguerre expansion form can prove less compact in practice than use of the discrete kernel form, *i.e.*, the distinct kernel values $h_0, h_1(i_1), h_2(i_1,i_2)$, $i_1 = 0, \ldots, 15, i_2 = 0, \ldots, i_1$. Another instance in which use of Laguerre expansion may prove less compact (resulting in lower estimation accuracy and increased computational requirement) is when the system kernels have an initial delay. An example is the Wiener model with linear filter

$$k(j) = j - 3, j = 4, \ldots, 16$$
$$= 0, \text{elsewhere},$$

followed by the polynomial $(\cdot) + (\cdot)^2$. In our attempt, using the criterion in Reference 57 to choose the rate of decay of the Laguerre functions did not yield an accurate result when 17 Laguerre functions were used (a number equal to the system memory length).

The use of Laguerre functions to estimate the kernels may be economical in instances when a system's kernels have smooth forms. In general, however, there is no *a priori* reason to assume that a system's kernels are smooth, and much detail provided by the distinct kernel values may be lost by an identification procedure that produces smoothed kernels.

An example is furnished by the kernels measured in (44) for two cells in the feline visual cortex. For each cell, an orthogonal method (45) was used to obtain two inde-

pendent estimates of kernels up to second order by using two distinct 1,000-point input segments. Each of the second-order kernel estimates had a very jagged appearance that could easily have been dismissed as a result of noise. In fact, however, the two estimates of the second-order kernel for each cell were virtually superimposable and had excellent replication of the jaggedness (but differed between the cells). In this application, a few Laguerre functions would not have sufficed to capture the jagged appearance but, rather, would have yielded much smoother kernels, that could erroneously have been considered to be more accurate.

It may be argued that simply increasing the number of Laguerre functions used to approximate the second-order kernel could enable one to recover more and more of the jagged shape. This is not necessarily true, because, as seen above, numerical difficulties may cause the Laguerre expansion approximation to be inaccurate even if the number of Laguerre functions used is equal to the memory length.

It has been claimed that, for systems with lengthy memory, use of Laguerre expansions to estimate kernels is less computer-memory intensive than are other methods. But this means that significantly fewer Laguerre functions will be used than the assumed memory length. Therefore, the comparison is not a fair one, because, in general, smoothed kernels will result; and "smoothed" may mean "less detailed," just as a smoothed version of an electrocardiogram has less information than the jagged original. Moreover, the smoothed kernels may well result in a significantly larger mean-square error of fit than will the kernels found by the orthogonal (45) or fast orthogonal (38,43) methods. Kernels with lengthy memory, for example, those with more than 11,000 distinct kernel values, can be recovered accurately (without introducing smoothing) via parallel cascade identification (43) on an ordinary personal computer (*e.g.*, limited to 640 kilo-bytes of RAM memory).

Ogura (61) suggests that systems in which the kernels exhibit an initial delay can be better represented by using associated Laguerre functions (see Eq. 167 in 61) and, therefore, he developed a fast algorithm to generate the outputs of biorthogonal Laguerre filters. Of course, the Watanabe-Stark (96) method of estimating the coefficients in the expansion by least-squares fitting over the data record also can be used here for increased accuracy.

Investigators (*e.g.*, 48,54,55,67–75) have made extensive use of Wiener kernel analysis to study the neural network of the catfish retina. This work, which was pioneered by Marmarelis and Naka (54), has revealed much detailed information about neural processing and neural connectivity in the vertebrate retina.

Note that even with a white-Gaussian input process, the higher-order orthogonal functionals become increasingly complicated. These functionals, however, are considerably more simplified than are the expressions that would be obtained if an attempt were made to calculate the corresponding orthogonal functionals for an input lacking suitable autocorrelation and probability density properties. The complexity derives from attempting to calculate the orthogonal functionals explicitly in terms of the input process, as in the Wiener G-functionals shown above. This complexity can be avoided by defining each orthogonal functional recursively in terms of previously created orthogonal functionals, as commonly done using Gram-Schmidt orthogonalization in many applications, such as functional expansions (see 62).

Sutter (89,90) has devised a powerful and very rapid procedure for kernel estimation that involves constructing multilevel test inputs using binary M sequences. Sutter (90) has noted that his procedure has an interesting analogy to the sum of sinusoids technique of Victor and Knight (93): Both methods use a test input constructed "to achieve exact orthogonality of the kernels and kernel estimates." Sutter's procedure is particularly valuable for analyzing nonlinear multi-input systems, such as the visual system (91), when each experimental input individually can be manipulated to follow precisely a multilevel sequence. Indeed, in such cases, Sutter's procedure may be the method of choice. When inputs cannot be so manipulated or when they merely can be monitored, Sutter (90) suggests that they may be "suitably parameterized for application of the binary algorithms," but this would introduce unnecessary measurement error. In the latter cases, alternate kernel estimation techniques, such as the fast orthogonal algorithm (38,43), parallel cascade identification (43), or the Watanabe-Stark method (96), may be indicated.

Exact Orthogonalization

In the orthogonal method for estimating the kernels accurately (45), recursively defined functions are created to be mutually orthogonal over the data record (or the portion of the record exceeding the system memory duration). Once the orthogonal functions have been created, their weightings can be determined simply via cross-correlations with the system output. The weightings in the orthogonal series then can be efficiently converted into the kernel values in the original Volterra series used to approximate the system. Note that the orthogonalization here occurs with respect to the particular input signal that evoked the measured output and is not independent of the input process. Because the functions used were made orthogonal over the data record, greatly improved kernel estimates resulted compared with estimates made with the Lee-Schetzen (51) approach to estimating the Wiener kernels, in which, theoretically, an ideal Gaussian input must be applied to the system for an infinite duration of time.

To illustrate the orthogonal method (45), consider the discrete-time second-order Volterra series

$$y_s(n) = h_0 + \sum_{i=0}^{I} h_1(i)\, x(n - i) + \sum_{i_1=0}^{I}\sum_{i_2=0}^{I} h_2(i_1,i_2)\, x(n - i_1)x(n - i_2), \quad (16)$$

let

$$y(n) = y_s(n) + e(n)$$

where x and y are respectively, input and output of the nonlinear system, e is the model error, and the kernels h_i are to be estimated. Suppose that the finite-length data record is defined for time instants $n = 0, \ldots, N$.

In simpler notation, Eq. 16 can be written equivalently as

$$y(n) = \sum_{m=0}^{M} a_m p_m(n) + e(n) \quad (17)$$

where $p_0(n)$ is unity and each $p_m(n)$, $m \geq 1$, is either an x term of the form $x(n - i)$ or a cross-product of x terms (*i.e.*, of the form $x(n - i_1)x(n - i_2)$). The a_m correspond directly to the individual kernel values to be estimated. The upper limit M in Eq. 17 is the total number (minus one) of distinct terms on the right side of Eq. 16. In more detail

$$\begin{aligned} p_0(n) &= 1 \\ p_m(n) &= x(n - m + 1) \quad m = 1, \ldots, I + 1 \end{aligned} \quad (18)$$

The line above produces the x terms in the first-order Volterra functional in Eq. 16. The cross-product (xx) terms in the second-order Volterra functional in Eq. 16 can be generated according to the following scheme:

```
m = I + 1
FOR i1 = 0 TO I
 FOR i2 = i1 TO I
  m = m + 1
  FOR n = I TO N
   pm(n) = x(n - i1) x(n - i2)
  NEXT n
 NEXT i2
NEXT i1
```

The $p_m(n)$ would be defined analogously to correspond to cross-product x terms in higher-order Volterra functionals when they are present.

Below, we shall create orthogonal functions $\beta_m(n)$ (via the Gram-Schmidt procedure) that correspond to distinct $p_m(n)$ above. These functions will be orthogonal for the actual input values over the record from $n = I$ to $n = N$, so that there is no theoretical requirement for an infinite record length or particular joint density functions for the input process. Of course, an input sufficiently rich in both amplitudes and frequencies is required for any identification technique to work. This richness of the input is reflected in its fractal dimension, and Victor (92) has shown how the fractal dimension (specifically, the capacity dimension) limits the number of terms identifiable in an orthogonal functional expansion representation.

Our objective is to rearrange Eq. 17 to have the form

$$y(n) = \sum_{m=0}^{M} g_m \beta_m(n) + e(n) \quad (19)$$

where the $\beta_m(n)$ are mutually orthogonal over the portion of the record from $n = I$ to $n = N$.

Following Wiener, we again use the Gram-Schmidt procedure to construct the orthogonal terms, but herein orthogonality is over the finite interval $[I, N]$ (as opposed to a theoretical infinite-length record). Moreover, herein the orthogonal $\beta_m(n)$ are defined recursively (as commonly done with the use of the Gram-Schmidt procedure) to avoid increasingly complex expressions. Finally, reconversion to the original Volterra series approximation is performed efficiently by a simple formula.

The modified Gram-Schmidt procedure (65) will be used to construct the orthogonal $\beta_m(n)$ from the defined $p_m(n)$, $m = 0, \ldots, M$ and $n = I, \ldots, N$. To understand the procedure, let each $p_m(n)$, $n = I, \ldots, N$, correspond to a vector of dimension $N - I + 1$, so that there are a total of $M + 1$ such vectors. The procedure begins by selecting one of the vectors, *e.g.*, $p_0(n)$, to be the first of the orthogonal vectors. Then the part of each remaining vector that is parallel to $p_0(n)$ is removed, to form the vectors $p_m^{(1)}(n)$, ($m = 1, \ldots, M$), each of which is orthogonal to $p_0(n)$. One of the $p_m^{(1)}(n)$, *e.g.*, $p_1^{(1)}(n)$, is selected to be the second orthogonal vector. The part of each remaining vector $p_m^{(1)}(n)$ that is parallel to $p_1^{(1)}(n)$ is removed to form the vectors $p_m^{(2)}(n)$, $m = 2, \ldots, M$. Each of the $p_m^{(2)}(n)$ is orthogonal to both $p_0(n)$ and $p_1^{(1)}(n)$. One of the $p_m^{(2)}(n)$, e.g., $p_2^{(2)}(n)$, is selected as the third orthogonal vector, and so on. Although the procedure is equivalent mathematically to the classical Gram-Schmidt procedure, this modified method (65) is much more numerically robust.

In more detail, for $n = I, \ldots, N$ and for $m = 0, 1, 2, \ldots, M$ let

$$p_m^{(0)}(n) = p_m(n). \quad (20)$$

By using the modified Gram-Schmidt procedure (65), for $r = 0, \ldots, M - 1$ and $m = r + 1, \ldots, M$, we define (37)

$$p_m^{(r+1)}(n) = p_m^{(r)}(n) - \alpha_{mr} p_r^{(r)}(n)$$

where the orthogonal functions are the

$$\beta_r(n) = p_r^{(r)}(n)$$

and

$$\alpha_{mr} = \frac{\overline{p_m^{(r)}(n)\ \beta_r(n)}}{\overline{\beta_r^2(n)}}. \tag{21}$$

Here, the overbar denotes the time average over the record portion from $n = I$ to $n = N$. For example,

$$\overline{p_m^{(r)}(n)\ \beta_r(n)} = \frac{1}{N - I + 1} \sum_{n=I}^{N} p_m^{(r)}(n)\ \beta_r(n).$$

Then it follows from Eq. 19 and the orthogonality of the $\beta_m(n)$ that

$$g_m = \frac{\overline{y(n)\ \beta_m(n)}}{\overline{\beta_m^2(n)}} \tag{22}$$

An alternative formula (37), based on the residue of $y(n)$ (rather than $y(n)$ directly) is

$$g_m = \frac{\overline{\left[y(n) - \sum_{j=0}^{m-1} g_j \beta_j(n) \right] \beta_m(n)}}{\overline{\beta_m^2(n)}}. \tag{23}$$

Once the g_m have been determined, then the system has been "identified" (more precisely, least squares fit) by the orthogonal series expansion on the right side of Eq. 19. The next step is to convert the g_m into the coefficients a_m (corresponding to the individual Volterra kernel values) on the right side of Eq.17. For this purpose, the following formula (45) can be used:

$$a_m = \sum_{i=m}^{M} g_i v_i \tag{24}$$

where

$$\begin{aligned} v_m &= 1 \\ v_i &= -\sum_{r=m}^{i-1} \alpha_{ir} v_r \quad \text{for } i = m + 1, \ldots, M \end{aligned} \tag{25}$$

It is clear that, as noted, there will be a direct correspondence between the required Volterra kernel values h_0, $h_1(i)$, $h_2(i_1,i_2)$, . . . and the coefficients a_m. This enables the kernel values to be written down directly from the values for the a_m. Therefore, the zero-order kernel is $h_0 = a_0$.

The first-order kernel is $h_1(i) = a_{i+1}$, $i = 0, \ldots, I$. The second-order kernel $h_2(i_1,i_2)$ can be recovered by using the following scheme:

```
m = I + 1
FOR i1 = O TO I
 FOR i2 = i1 TO I
  m = m + 1
  h2(i1,i2) = am
  IF i1 ≠ i2 THEN h2(i1,i2) = 0.5 h2(i1,i2)
 NEXT i2
NEXT i1
```

Similar formulas for estimating the higher-order kernels can be set down analogously. Fast orthogonal methods for conducting the kernel estimation also are available (38,40,42,43).

Advantages, Disadvantages, and Further Details of the Orthogonal Method (45)

1. For any assumed order, the Volterra kernel expansion obtained by this method results in the minimal mean square error of all functional expansions of that order for the input and duration used in the identification experiment.
2. Wiener kernel values can be identified simply by use of a white-Gaussian input, because a finite-order Wiener series (47) that contains the Wiener kernels (which are input-variance dependent) can be rearranged into a corresponding Volterra expansion that minimizes the mean-square error for such an order and input. For example, suppose a system can be represented exactly by an infinite-order Volterra functional expansion but is approximated by a second-order Volterra series with a white-Gaussian input. Then, the first- and second-order kernels obtained by the orthogonal method are, in fact, the Wiener kernels that correspond to that input. If the approximating series were of the i-th order, then the i-th and $(i - 1)$-th order kernels obtained would equal the Wiener kernels of corresponding order. (If a colored-Gaussian input were used, then the Fourier-Hermite generalizations of the Wiener kernels would be obtained.)
3. The addition of further terms to the orthogonal expansion will not affect the values of the orthogonal terms previously determined.
4. An implementation of this method should eliminate those $\beta_m(n)$ for which $\overline{\beta_m^2(n)}$ is less than some specified small value. This helps to avoid division by a very small number in Eqs. 21 and 22, which would result in significant inaccuracies in the final estimates.
5. In contradistinction to the Lee-Schetzen method, the

diagonal values of the kernels can be found with equal accuracy as can the off-diagonal values without modification of the orthogonal method examined above.

6. If the same input is to be applied repeatedly in various experiments, then it is only necessary to generate and store the orthogonal functions $\beta_m(n)$ and the α_{mr} once. Under these conditions, the orthogonal method execution time can be reduced even further.
7. The method avoids the need for lengthy inputs. This is because the orthogonalization is over the actual data record, *i.e.*, specified for the actual input used. In many experimental environments, short inputs not only are desirable, but sometimes are mandatory (*e.g.*, preparation fatigues or dies). The fact that short input records suffice for the orthogonal method makes it possible to subdivide the total record length into smaller segments for individual identification and thereby, verification of the kernels obtained. This also enables slowly time-varying systems to be tracked by successive determination of snapshots of the kernels. Moreover, shorter record lengths reduce execution time for identification. Discussion of suitable record lengths is provided below in simulations of the related fast orthogonal algorithm (38,42,43).
8. The orthogonal method (45) also eliminates the requirement that the input have special autocorrelation (*e.g.*, white) or probability density (*e.g.*, Gaussian) properties. Frequently, it is impossible to produce white-Gaussian inputs because of both bandwidth and amplitude range limitations on the experimental apparatus.
9. Even when the input is an excellent white-Gaussian approximation, the method achieves estimates that are orders of magnitude more accurate than are those obtained by the cross-correlation method of Lee and Schetzen (51). Even when the input is 100 times longer, in our experience, the cross-correlation method does not achieve the same accuracy as does the orthogonal method with the shorter input.
10. A disadvantage of the orthogonal method in the above-reviewed form is that the execution time is many times longer than is that of the Lee-Schetzen cross-correlation technique, which may become significant when the memory length is large.
11. A fast orthogonal algorithm developed by Korenberg (38,42,43) eliminates the need to create the orthogonal functions explicitly and exploits the lagged nature of the input terms in the Volterra series, thereby reducing dramatically the computing time. This faster algorithm retains the advantages noted, in particular, the significant improvement in accuracy over the cross-correlation method. Moreover, the faster algorithm enables second-order Volterra series approximations with memory lengths greater than 50 and record lengths greater than 10,000 to be calculated within a few minutes on a modern microcomputer. This is illustrated in the following section, but it should be noted that third- and higher-order kernels also can be estimated by this method.

Simulation Results for Kernel Estimation

The simulated system consisted of a dynamic linear subsystem followed by a squarer with unity offset. The output, y, was given by

$$y(n) = \left(\sum_{i=0}^{I} h(i)\, x(n-i) \right)^2 + 1 + e(n) \qquad (26)$$

for $n = 0, 1, \ldots, N$

where x is the input, y is the output, $N + 1$ is the number of input-output data pairs, h is a (discrete) impulse response function, and e is zero-mean, Gaussian white noise (corrupting the output) with a specified variance. Note that $I + 1$ is the memory length of the system, because the output depends on input values that have delays from 0 to I lags. The impulse response function, h, was obtained by sampling (at 0.05-second increments) a continuous low-pass second-order impulse response function, the Laplace transform of which was

$$H(s) = \frac{a\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

where $\omega_n = 6$, $\zeta = 0.2$, and $a = 2$.

In the simulation, the input x was approximately zero-mean, unity variance, Gaussian white noise. Kernel estimation by means of the fast orthogonal algorithm of Korenberg (38,42,43) and the cross-correlation method of Lee and Schetzen (51) was investigated in both noise-free and noisy conditions. A second-order Volterra series was used as the test system in this example, because, in that case, the corresponding Wiener and Volterra kernels of first and second-order are equal, as observed above and discussed further below. The fast orthogonal algorithm (38,43) does not require that the input be Gaussian white noise: This input was used here to enable use of the Lee-Schetzen (51) technique.

In the first study, the memory length $(I + 1)$ equaled 55. Therefore, 1 zeroth-order, 55 first-order, and 1,540 distinct second-order kernel values were to be estimated by both the cross-correlation method and the fast orthogonal algorithm with the use of 5,000 input-output data pairs. In general, the data record is assumed to be available for $n = 0, \ldots, N$, but the Volterra series is best-fit over only $n = I, \ldots N$. Therefore, to estimate a second-order series requires at least $1 + I + 1 + (I + 1)(I + 2)/2 + I$ (here 1,650) data

pairs, but this would allow no redundancy. Because algorithm performance in a noisy environment is examined below, the number of data pairs used was approximately three times the minimum required.

The actual first-order kernel is zero for the test system and, indeed, was found to be negligible to six significant digits by the fast orthogonal algorithm. Figure 1 shows the actual second-order kernel of the test system plotted together with the kernel estimated using the fast orthogonal algorithm. The two plots are indistinguishable because the fast orthogonal kernel estimate was accurate to six significant digits. For comparison, Fig. 2 shows the second-order kernel estimated via cross-correlation. It is clear that the fast orthogonal algorithm is much more accurate. Indeed, the estimates obtained here by this method are 7 orders of magnitude more accurate than are those obtained via cross-correlation. The kernel estimates shown in Figs. 1 and 2 were obtained by using noise-free data. Figure 2 is shown after 3 × 3 smoothing, *i.e.*, the second-order kernel value at lag pair i_1, i_2 is found by averaging the nine kernel values corresponding to a 3 × 3 grid centered at i_1, i_2. This smoothing is beneficial when it is known that the system kernels have smooth shapes. For such kernels, use of a basis, such as the Laguerre functions, to expand the kernels will result in smooth kernel estimates even with noisy data. This is because the basis functions themselves are smooth and may economically fit certain smooth kernel shapes. Therefore, by choosing the system kernels to be smooth, one can engineer a test in which use of Laguerre functions apparently can give increased accuracy on noisy data. However, the use of mild smoothing (such as over nine adjacent points) can be equally effective (see below).

Figures 3 and 4 (3 × 3 smoothed) show the effect on fast orthogonal estimates of corrupting the output with

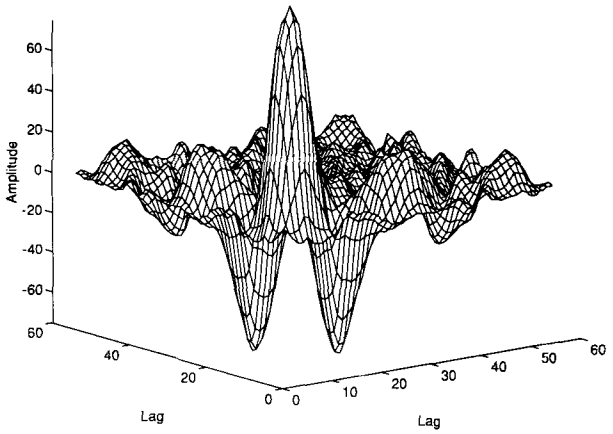

FIGURE 2. The second-order kernel (3 × 3 smoothed) estimated by using the cross-correlation method, for a memory length of 55 and a noise-free record of 5,000 data pairs.

small and large amounts of additive noise ($e(n)$ in Eq. 26). Figure 3 is the second-order kernel estimate obtained when the variance of the corrupting noise was 5% of the variance of the noise-free output. Figure 4 shows the second-order kernel estimate for a 50% noise corruption. It should be noted that this kernel estimate is very close to the actual kernel (Fig. 1) and much more accurate than is the cross-correlation estimate (Fig. 2) obtained under noise-free conditions.

In the second study, the effects of changes in memory length ($I + 1$) and record length ($N + 1$) were investigated, together with an estimation of computing time. Figure 5 shows the measured total time taken to estimate the zeroth-, first-, and second-order Volterra kernels by using the fast orthogonal algorithm implemented in the C language running on an IBM RISC/6000 320H workstation. The number of input samples ($N + 1$) was fixed at 5,000, and the memory length ($I + 1$) varied from 10 to 55. Note

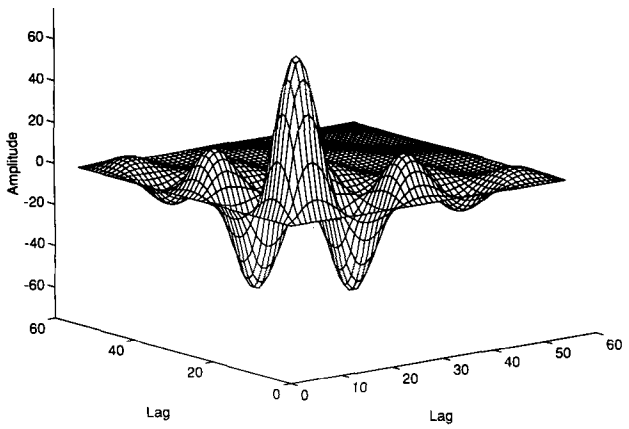

FIGURE 1. Actual second-order kernel of the test system plotted together with the kernel estimated by using the fast orthogonal algorithm, for a memory length of 55 and a noise-free record of 5,000 data pairs. The two plots are indistinguishable.

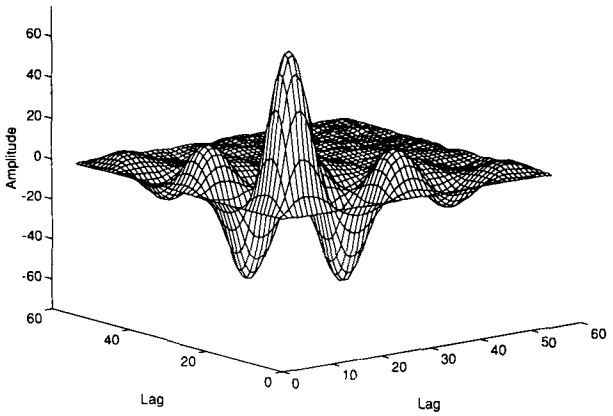

FIGURE 3. The second-order kernel (3 × 3 smoothed) estimated by using the fast orthogonal algorithm, for a memory length of 55 and a noisy record of 5,000 data pairs. Here, the noise corrupting the output had a variance equal to 5% of the noise-free output variance.

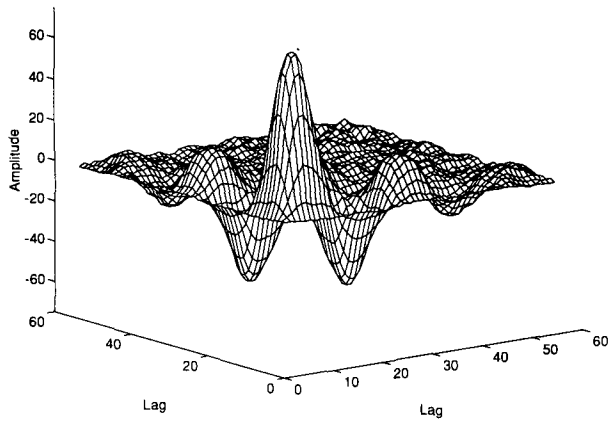

FIGURE 4. The second-order kernel (3 × 3 smoothed) estimated by using the fast orthogonal algorithm, for a memory length of 55 and a noisy record of 5,000 data pairs. Here, the noise corrupting the output had a variance equal to 50% of the noise-free output variance. This should be compared with Fig. 2, which shows the performance of the cross-correlation method with the use of noise-free data.

that the logarithm of the execution time increases approximately linearly with the memory length. A memory length of 55 (corresponding to the estimation of 1,596 distinct kernel values) required an execution time of 1,600 seconds. Sutter's method (89,90) would require only a fraction of this time, if it were possible to apply precisely a suitable multilevel input to the system. If this is not possible, then using the fast orthogonal algorithm (38,43) or obtaining kernels via parallel cascade identification (43) may well be indicated to provide high accuracy.

Figure 6 shows the execution times for the fast orthogonal algorithm with the use the same computer and code, when the memory length ($I + 1$) was held constant at 40 and the number of samples ($N + 1$) was varied from 1,000 to 100,000. Each simulation involved the estimation of 861 distinct kernel values. Note that the execution time increases nearly linearly with the number of samples. For comparison, singular value decomposition (SVD algorithm *sydcmp* in 64) was used to solve for least-square estimates of the kernel values. For example, for a memory length of 35, the total running time, starting with the raw input-output data, was 36 times longer than was that of the fast orthogonal algorithm.

IDENTIFICATION OF CASCADES (BLOCK-STRUCTURED MODELS)

The more precise kernel identification achieved by orthogonalizing over the actual data record (38,42,43,45) enables more accurate estimation of block-structured models. Consider, for example, the discrete two-input two-output cascade of Fig. 7, in which the square boxes denote the filters of dynamic linear subsystems and the rectangular boxes denote two-dimensional static nonlinearities. Suppose that the inputs x_1 and x_2 are mutually independent Gaussian white processes. In this case, it has been shown (36) that the cross-correlations

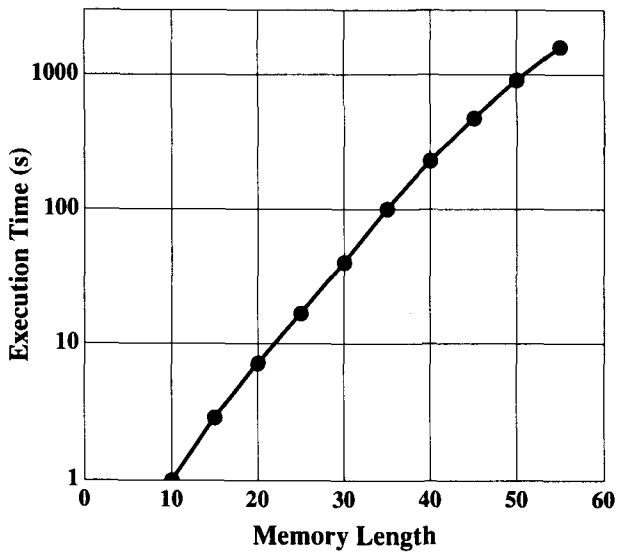

FIGURE 5. The measured times taken to estimate all of the zeroth-, first-, and second-order kernels in a second-order Volterra series approximation for a nonlinear system. The times are for the use of the fast orthogonal algorithm and are plotted as a function of the kernel memory length. Here, 5,000 input-output data pairs were used. The algorithm was coded in C and run on an IBM RISC/6000 320H workstation.

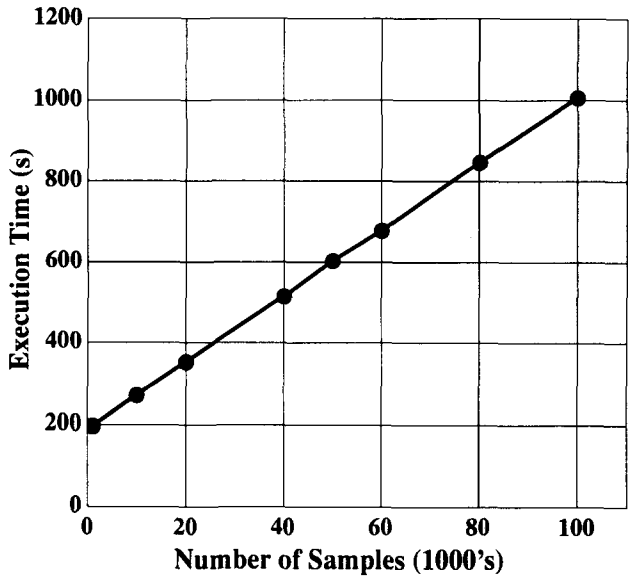

FIGURE 6. The measured times taken to estimate all of the zeroth-, first-, and second-order kernels by using the fast orthogonal algorithm plotted as a function of the number of input-output data pairs used. The kernel memory length was 40, corresponding to 861 distinct kernel values. The algorithm was coded in C and run on an IBM RISC/6000 320H workstation.

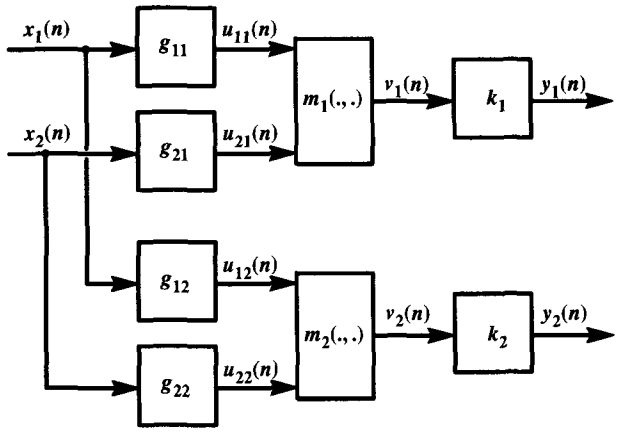

FIGURE 7. Two-input two-output LNL cascade. The square boxes denote dynamic linear systems and the remaining rectangular boxes denote two-input single-output static nonlinearities.

$$\phi_{xy_j}(n_1) = E\{y_j(n)x_i(n - n_1)\}$$

$$= A_{ij} \sum_{r=0}^{\infty} k_j(r)\, g_{ij}(n_1 - r) \qquad (27)$$

$$\phi_{xxy_j}(n_1,n_2) = E\{(y_j(n) - E\{y_j(n)\})\, x_i(n - n_1)\, x_i(n - n_2)\}$$

$$= A_{iij} \sum_{r=0}^{\infty} k_j(r)\, g_{ij}(n_1 - r)\, g_{ij}(n_2 - r) \qquad (28)$$

where A_{ij} and A_{iij} are constants, and $E\{.\}$ denotes infinite time-average taken over n. Here, $i = 1, 2$ (two inputs) and $j = 1, 2$ (two outputs), but these results hold for analogous multivariable cascades with arbitrary numbers of inputs and outputs (36).

The basic results for the SISO cascade of a dynamic linear, a static nonlinear, and a dynamic linear subsystem, frequently called an LNL cascade (46), can be shown (36) to carry over to such multi-input multi-output (MIMO) cascades. For example, it is widely known (11,82) that, for the SISO cascade, the first-order Wiener kernel is proportional to the convolution of the two dynamic linear subsystems ("sandwiching" the static nonlinearity), and Eq. 27 is a generalization of this result. Similarly, Eq. 28 is a MIMO generalization of the result (32,33) for the second-order cross-correlation of an SISO LNL cascade. In addition it has been shown that the dynamic linear subsystems of the SISO cascade can be estimated readily, from the cross-correlation functions, in both the frequency domain (32) and the time domain (34,36). Moreover, the Wiener kernels for the cascade are proportional to the Volterra kernels of corresponding order (32,33), provided that the cascade has a Volterra series representation as well as a Wiener functional expansion. (The Volterra series will exist when the static nonlinearity between the dynamic linear subsystems is representable by a polynomial or a power series.) The shape of the dynamic linear filter preceding the static nonlinearity is given by the first nonnegligible slice of the second-order (symmetric) Wiener kernel parallel to one axis (34,36). Multiple slices of the second-order Wiener kernel also can be used (39) to give improved estimates of the first dynamic linear subsystem in the LNL cascade. The same results apply and can be used to identify MIMO cascades (36) analogous to that in Fig. 7, and, for this reason, such cascades properly can be called MIMO LNL cascades.

In Fig. 7, suppose that x_1 and x_2 are mutually independent, Gaussian white processes applied simultaneously. Assume that we measure the first- and second-order Wiener kernels $w_1(j)$ and $w_2(j_1,j_2)$ when x_1 is taken as the input and y_1 as the output. Then (34,36), the shape of the dynamic linear filter $g_{11}(j)$ is proportional to the first nonnegligible term of the sequence $w_2(j,0)$, $w_2(j,1)$, ... Next, it follows from Eq. 27, that the first-order Wiener kernel $w_1(j)$ is proportional to the convolution of g_{11} and k_1. Therefore, the filter $k_1(j)$ can be obtained, up to arbitrary scaling constant and horizontal shift, by deconvolving g_{11} from $w_1(j)$. Alternatively, the identified filter g_{11} can be used to calculate $u_{11}(n)$ (again, up to an arbitrary proportionality constant and time shift). The system with u_{11} as input and y_1 as output is a Hammerstein model (26), the first-order kernel and second-order kernel diagonal values of which each are proportional to $k_1(j)$. Therefore, $k_1(j)$ can be determined up to a proportionality constant and horizontal shift.

Similarly, we can obtain g_{21}, and a second estimate of k_1, by measuring the first- and second-order Wiener kernels when x_2 is taken as input and y_1 as output. When g_{11}, g_{21}, and k_1 are known, the static nonlinear characteristic $m_1(.,.)$ can be estimated by a simple least-squares procedure with the use of calculated values of u_{11} and u_{21} and the known output y_1. The filters g_{12}, g_{22}, and k_2 and static nonlinear characteristic $m_2(.,.)$ can be estimated analogously.

Another, usually more accurate, alternative for estimating the dynamic linear filters is to apply each of the Gaussian inputs separately. For example, when x_1 is the sole input, the SISO cascade enables g_{12} and k_2 estimates when y_2 is the output and g_{11} and k_1 estimates when y_1 is the output. Analogously, having x_2 as sole input enables estimates of g_{21} and g_{22} and reestimates of k_1 and k_2 to be obtained.

It will be appreciated that the two-input two-output cascade of Fig. 7 can be treated as a pair of two-input

single-output cascades. Accordingly it is sufficient to illustrate identification of such simpler cascades.

Simulation Results for MIMO Cascade Identification

Figure 8 shows a two-input single-output cascade. Only the identification of the dynamic linear filters is illustrated, because the estimation of the two-dimensional static nonlinearity is then straightforward. Note that, although the defining polynomial for the static nonlinearity is indicated in Figs. 8 and 10, the static nonlinearity was not assumed to be known in identifying the cascade.

Two different experimental paradigms were used to compare the accuracy of the filter estimates. First, filter estimates (in Fig. 8) were obtained when x_1 and x_2 were simultaneously applied, 1,000-point, white Gaussian sequences that were approximately mutually independent (normalized cross-correlation shown in Fig. 9). (These inputs were created by using an extension of a noise-generation technique proposed in 24.) Second, the same inputs x_1 and x_2 were applied individually, resulting in the filter estimates shown in Fig. 10. In both paradigms, each linear filter preceding the static nonlinearity was estimated from the first nonnegligible slice, parallel to one axis, of the relevant second-order Wiener kernel (34,36), *i.e.*, the first nonnegligible term of the sequence $w_2(j,0), w_2(j,1), \ldots$. The first- and second-order Wiener kernels were estimated by approximating the relevant input output data with a second-order Volterra series and by using the fast orthogonal algorithm (38,42,43).

In Figs. 8 and 10 the cascade inputs x_1 and x_2, respectively, had means of -0.001 and -0.002, and variances of 1.000 and 0.996. The signals u_1 and u_2 had variances of 1,391 and 1,306, respectively. Because only one of the simultaneously applied x_1, x_2 is taken as input (with y as output) in calculating the Wiener kernels, at the input to the static nonlinearity (in Fig. 8) there is a signal-to-noise ratio of approximately unity. This explains why the filter

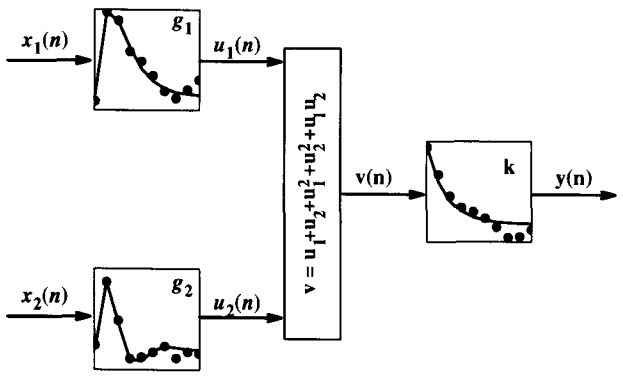

FIGURE 8. Two-input single-output LNL cascade forming the simulated nonlinear system to be identified. The square boxes show the true (solid line) impulse responses of the dynamic linear systems together with the estimated values (points).

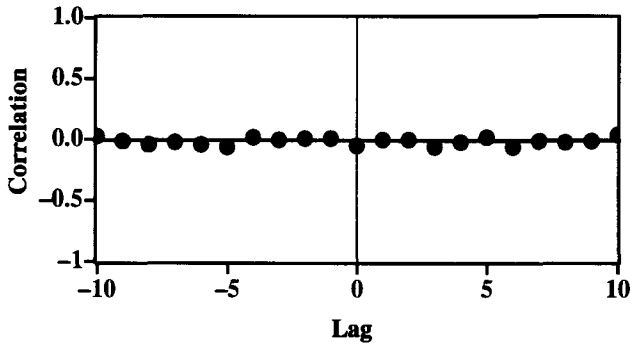

FIGURE 9. Cross-correlation of the Gaussian white inputs x_1 and x_2 simultaneously applied to the LNL cascade of Fig. 8 to identify it. The graph shows that the two inputs are nearly uncorrelated.

estimates in Fig. 10, in which the inputs x_1 and x_2 are applied individually, are superior to the estimates in Fig. 8, which result from simultaneously applied inputs. However, the Fig. 8 filter estimates are surprisingly accurate given the poor signal-to-noise ratio. The estimates can be improved by increasing the experimental duration, by 3×3 smoothing the second-order kernels, and by using input signals (for x_1, x_2) for which the normalized cross-correlation function is closer to zero than is that shown in Fig. 9.

Finally, an iterative scheme can be used to increase the accuracy of the filter estimates for both SISO and MIMO cascades. To illustrate, in Fig. 10, begin by estimating g_1 from the first nonnegligible slice through the second-order Wiener kernel parallel to one axis. With the use of calculated values of $u_1(n)$ as input and the known $y(n)$ as output, measure the first- and second-order Wiener kernels and estimate k from the first-order or second-order diagonal kernel values (or both). Then, calculate the two-sided inverse (25) of k, and use this inverse, with $y(n)$ as input, to estimate $v(n)$. The nonlinear system with x_1 as input and $v(n)$ as output is a Wiener model (26) (not to be confused with a Wiener series). This model's first-order Wiener

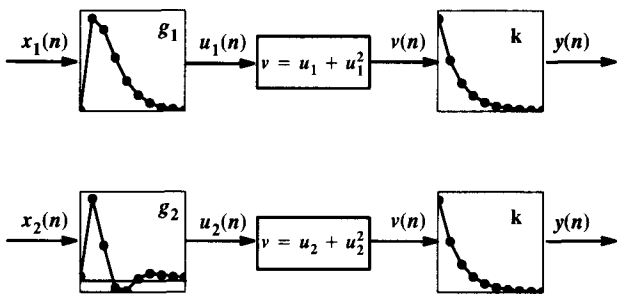

FIGURE 10. The two single-input single-output LNL cascades resulting from Fig. 8 by applying individually the inputs x_1 and x_2. The square boxes show the true (solid line) impulse responses of the dynamic linear systems together with the estimated values (points).

kernel and *any* slice of the second-order kernel parallel to one axis is proportional to g_1. Therefore, calculate these Wiener kernels using x_1 as input and the estimated $v(n)$ as output, and then use the first-order kernel, and/or slices (or, preferably, the sum of slices) of the second-order kernel, to reestimate g_1. A new estimate of u_1 can now be calculated, and so forth.

RELATION OF VOLTERRA TO WIENER KERNELS

The Wiener kernels, in general, are not the same as the Volterra kernels of corresponding order. For example, the zeroth-order Volterra kernel simply is the system output when the system input is zero, whereas the zeroth-order Wiener kernel is the mean output for the particular Gaussian white input used. However, a straightforward relation between Volterra and Wiener kernels exists, as does a relation between Wiener kernels measured with different input mean levels or power densities (27–29,56). Suppose that the Wiener kernels are measured with zero input mean. To calculate the i-th-order Volterra kernel from the Wiener kernels requires knowing the Wiener kernels of order i, $i + 2$, . . . (an analogous requirement exists when converting from Volterra to Wiener kernels). If a system is described by a finite-order (*e.g.*, Q) Volterra series, as on the right side of Eq. 5 when $\epsilon(t) = 0$, then the Volterra kernels of order $Q - 1$ and Q will equal the corresponding Wiener kernels. Therefore, the corresponding first- and second-order Volterra and Wiener kernels are identical when the system has no higher-order kernels.

APPLICATIONS OF THE VOLTERRA KERNEL APPROACH

Perhaps the first to use the Volterra kernel representation in the biological sciences was Stark (see 77,83–85), who modeled the human pupillary control system. Stark assumed that the relation between light flux as input and pupil area as output could be approximated by the first two terms of a Volterra expansion (85). He then (84,85) obtained first- and second-order kernels by using two different procedures. First, by using noise excitation experiments, he found two Wiener kernels that would best characterize the system in the mean-square sense. Assuming that the system is only second-order, these also are the Volterra kernels. Second, by using Schetzen's (78) double-pulse experiments, Stark found the two Volterra kernels which would completely fit the resultant data. The poor agreement between the second-order kernels found by these two procedures was attributed to the "use of different excitation functions—random noise and multiple pulses—[that] puts the complex pupillary system into different operating conditions yielding the different respective kernels" (85).

Another possible explanation is that the pupillary system cannot be approximated accurately by only the first two terms of the Volterra expansion for the inputs used by Stark, and, therefore, neither of the two procedures provided reliable estimates of the Volterra kernels. In other words, if Stark's approximation of the system had contained more terms of the Volterra expansion, then this well might have changed the expressions obtained for the first- and second-order Volterra kernels.

Krenz and Stark (49) provide an excellent discussion of kernel approaches to the pupillary control system, including some issues raised herein. Hung and Stark (23) have considered the relation between the internal structure of a system and the shape of the system kernels, in particular, in connection with the human pupillary system. More generally, the relation between system structure and kernels is examined in References 12, 26, 32–36, 46, and 47.

By using a kernel estimation technique based on Volterra's original definition of kernels, Korenberg (31) investigated the phenomenon of pupillary contraction after both light and dark brief pulses of light (14). Korenberg (31) showed that, if the pupil exhibits a contractile response for a fixed interval after any positive or negative brief pulse of light, then the first-order Volterra kernel is identically zero in this interval and the second-order kernel is nonpositive along its diagonal. This, of course, assumes that the pupillary control system can indeed be represented by a Volterra series.

Another approach to dealing with system nonlinearities is the use of the describing function (81), which involves expressing the system frequency response as a function of input amplitude. This approach has been used successfully by Stark *et al.* (86) to analyze a representation of the human lens control system having a Hammerstein model in the forward loop.

A very interesting application of the kernel approach was made by Litt (52), who studied medical imaging techniques, such as magnetic resonance imaging (MRI) and computed tomography (CT). Considering them to be nonlinear systems, he used the fast orthogonal algorithm (38,43) to calculate kernels for four cases: 1) photo to CT; 2) photo to MRI; 3) CT to MRI; and 4) one MRI to another MRI system having different adjustments for some of the acquisition parameters. In the first case, for example, the input was a photograph of a two-dimensional phantom containing barium-loaded agarose, while the output was the CT image corresponding to the same phantom. In each of the cases, both the input and the output were two-dimensional spatial images obtained by an imaging device from the same phantom. Hence spatial rather than temporal kernels up to second-order were estimated for the nonlinear system transforming the input image into the output image. Since both input and output images were outputs of imaging devices which can introduce distortions, this was an example of when the input could be measured but not

in general manipulated to follow precisely a multi-level sequence. Thus the fast orthogonal algorithm (38,43), which yields accurate kernel estimates for a very wide class of inputs, proved advantageous in this application. Litt (52) found second-order nonlinearities "in both CT and MRI related to spatial factors such as geometric distortions and edge-effects, and grey map transformations between modalities."

In the case of comparing different MRI systems, Litt (52) for example estimated the zero-, first-, and second-order spatial kernels when the input image, obtained using a short repetition time and a single acquisition, was much noisier than the output image, obtained using a repetition time six times longer than that for the input image and four acquisitions. The estimated kernels then may be applied to new noisy images to transform them into better ones. In suggesting this method of improving the MR images, Litt (52) was motivated by the work of Sunay and Fahmy (88), who used fast orthogonal search, proposed by Korenberg (40), to obtain spatial nonlinear difference equations when the input image was a noise-corrupted version of the output image. The identified difference equations then served as two-dimensional nonlinear filters capable of dramatically reducing the noise corruption of both the original noisy image and new test images (88).

Another interesting application of the kernel approach was made by Marmarelis *et al.* (58), who modeled some renal data using a third-order Volterra series approximation. Here arterial blood pressure was considered to be the system input and renal blood flow the output. They used the Watanabe-Stark method (96) of estimating the kernels via least-square fitting Laguerre expansions and found that only a few Laguerre functions were needed in the third-order Volterra series approximation to achieve a low mean-square error. Indeed, the most important finding in Reference 58 likely is that the renal blood flow can be approximated with high accuracy as an explicit functional of the arterial blood pressure by using only a relatively few parameters or coefficients.

When Marmarelis *et al.* (58) used a second-order Volterra series approximation, only 45 parameters were used in expanding the kernels with Laguerre functions; they introduced 120 additional parameters in moving to the improved third-order series approximation (58). In the renal experiments, the record comprised 512 data pairs, whereas the assumed memory length (number of input lags) was 61. Therefore, a second-order Volterra series would have $1 + 61 + 61 \times 62/2 = 1{,}953$ distinct kernel values, whereas a third-order Volterra series would have 41,664. Therefore, in this application, the Watanabe-Stark method (96) of expanding the kernels by using Laguerre functions proved to be efficient, because a few Laguerre functions and, therefore, relatively few coefficients, were required for an excellent fit to the data.

One significant conclusion of Reference 58 is that the third-order kernel plays an important role in explaining certain "depressions" in renal blood flow. Although this may well be true, it is not, in our opinion, substantiated by the study. This is because a second-order Volterra series with a memory length of 61 (having 1,953 distinct kernel values) could reproduce *exactly* the 512 point input-output record. Had such a second-order series actually produced the experimental record, then any conclusion of the importance of the third-order kernel clearly would be fallacious.

The reason Marmarelis *et al.* (58) had to use a third-order series approximation for an improved fit may have been because of the very smooth kernels that result from using only a few Laguerre functions. If more detail were allowed in the first- and second-order kernels, rather than the smooth shapes presented in the study (58), then the exact input-output record could be accounted for by a second-order Volterra series. The essential point is that there simply were not enough data points available to uniquely specify even a second-order Volterra series with memory length 61. Therefore, there is no way to know how well the kernels (up to third order) in the study (58) resemble the actual system kernels.

Indeed, the paucity of data prevented Marmarelis *et al.* (58) from following the usual practice of splitting the record into a training set (for estimating kernels) and a testing set (for measuring mean-square error). Therefore, outputs from their obtained Volterra series approximations are not evoked by novel inputs and, therefore, are not real "predictions." It is not surprising, therefore, that the power spectrum calculated from the obtained third-order Volterra series model is indeed much closer to the actual output power spectral density than is that from a lower-order model. Far more parameters were used to fit the third-order model, and the power spectrum was not predicted for a new input, so the validity of the model remains open to question. In a subsequent study, Chon *et al.* (13) used fast orthogonal search (40) to fit the same renal data very concisely and accurately by nonlinear difference equations. **(Note added in proof)**

CONCLUSION

We have reviewed critically the use of Volterra series approaches in nonlinear biological and physiological systems, and have mentioned a recent application to medical imaging systems. We have referred to a number of techniques that can be used to estimate kernels, such as Schetzen's multipulse method, the Watanabe-Stark utilization of least-square fitting in expanding the kernels via Laguerre functions, Sutter's innovative approach using multilevel inputs constructed using binary M sequences, and Korenberg's parallel cascade identification method.

Some guidelines were provided to indicate when it may be appropriate to use a particular method for kernel estimation.

We have shown that a vast improvement of the accuracy in estimating the kernels in a Volterra series approximation for a system is attainable by using a technique (38,43) that implicitly transforms the Volterra series into a series that is orthogonal over the actual data record (or the portion exceeding the system's memory), determines the weightings of the orthogonal terms, and then rapidly converts these weightings through an efficient formula into the kernels in the original Volterra series.

REFERENCES

1. Alper, P., and D. Poortvliet. On the use of Volterra series representation and higher order impulse responses for nonlinear systems. *Elec. Lab. Rep. Tech. Univ.* Delft, Netherlands, 1963.
2. Andronikou, A. M., G. A. Bekey, and S. F. Masri. Identification of nonlinear hysteretic systems using random search. *IFAC Ident. Sys. Param. Est.* 1:263–268, 1982.
3. Barrett, J. F. The use of functionals in the analysis of nonlinear physical systems. *J. Elect. Control* 15:567–615, 1963.
4. Barrett, J. F. The use of Volterra series to find region of stability of a nonlinear differential equation. *Int. J. Control* 1:209–216, 1965.
5. Barrett, J. F. Functional series representation of nonlinear systems—some theoretical comments. *6th IFAC Symp. Ident. Sys. Param. Est.* 1:251–256, 1982.
6. Billings, S. A., and W. S. F. Voon. Correlation based model validity tests for nonlinear models. *Int. J. Control* 44:235–244, 1986.
7. Bose, A. G. A theory of nonlinear systems. *MIT Res. Lab. Elec. Tech. Rep.* 309, 1956.
8. Boyd, S., and L. O. Chua. Fading memory and the problem of approximating non-linear operators with Volterra series. *IEEE Trans. Circ. Sys.* 32:1150–1160, 1985.
9. Boyd, S., Y. S. Tang, and L. O. Chua. Measuring Volterra kernels. *IEEE Trans. Circ. Sys.* 30:571–577, 1983.
10. Brilliant, M. B. Theory of the analysis of nonlinear systems. *MIT Res. Lab. Elec. Tech. Rep.* 345, 1958.
11. Bussgang, J. J. Crosscorrelation functions of amplitude distorted Gaussian signals. *MIT Res. Lab. Elec. Tech. Rep.* 216:1–14, 1952.
12. Chen, H. W., L. D. Jacobson, and J. P. Gaske. Structural classification of multi-input nonlinear systems. *Biol. Cyber.* 63:341–357, 1990.
13. Chon, K. H., N. H. Holstein-Rathlou, D. J. Marsh, and V. Z. Marmarelis. Parametric and nonparametric nonlinear modeling of renal autoregulation dynamics. In: *Advanced Methods of Physiological System Modeling,* vol. 3. edited by V. Z. Marmarelis. New York: Plenum Press, 1994, pp. 195–210.
14. Clynes, M. The non-linear biological dynamics of unidirectional rate sensitivity illustrated by analog computer analysis, pupillary reflex to light and sound, and heart rate behavior. *Ann. NY Acad. Sci.* X:98, 1962.
15. Ewen, E. J., and D. D. Weiner. Identification of weakly nonlinear systems using input and output measurements. *IEEE Trans. Circ. Sys.* 27:1255–1261, 1980.
16. Flake, P. Volterra series representation of nonlinear systems. *AIEE Trans.* 64:330–335, 1963.
17. Frechet, M. Sur les fonctionnelles continues. *Annales Scientifiques de l'Ecole Normal Superieure* 27:193–219, 1910.
18. Gardner, W. A., and T. L. Archer. Exploitation of cyclostationarity for identifying the Volterra kernels of nonlinear systems. *IEEE Trans. Inform. Theory* 39:535–542, 1993.
19. George, D. Continuous nonlinear systems. *MIT Res. Lab. Elec. Tech. Rep.* 335:246–281, 1959.
20. Goussard, Y., W. C. Krenz, and L. W. Stark. An improvement of the Lee and Schetzen cross-correlation method. *IEEE Trans. Automat. Control* 30:895–898, 1985.
21. Goussard, Y., W. C. Krenz, L. W. Stark, and G. Demoment. Practical identification of functional expansions of nonlinear systems submitted to non-Gaussian inputs. *Ann. Biomed. Eng.* 19:401–427, 1991.
22. Hung, G., and L. Stark. The kernel identification methods: review of theory, calculation, application, and interpretation. *Math. Biosci.* 37:135–170, 1977.
23. Hung, G. K., and L. W. Stark. The interpretation of kernels: an overview. *Ann. Biomed. Eng.* 19:509–519, 1991.
24. Hunter, I. W., and R. E. Kearney. Generation of random sequences with jointly specified probability density and autocorrelation functions. *Biol. Cybern.* 47:141–146, 1983.
25. Hunter, I. W., and R. E. Kearney. Two-sided linear filter identification. *Med. & Biol. Eng. & Comput.* 21:203–209, 1983.
26. Hunter, I. W., and M. J. Korenberg. The identification of nonlinear biological systems: Wiener and Hammerstein cascade models. *Biol. Cybern.* 55:135–144, 1986.
27. Klein, S. Interpreting nonlinear systems: the third order kernel of the eye movement control system. In: *10th Annual Pittsburgh Conference on Modeling and Simulation,* edited by Vogt and Mickle. Pittsburgh, PA: University of Pittsburgh, 215–222, 1979.
28. Klein, S. A. Relationships between kernels measured with different stimuli. In: *Advanced Methods of Physiological System Modeling,* vol. I, edited by V. Z. Marmarelis. Los Angeles: Biomedical Simulations Resource, University of Southern California, 1987, pp. 278–288.
29. Klein, S., and S. Yasui. Nonlinear systems analysis with non-Gaussian white stimuli: general basis functionals and kernels. *IEEE Trans. Inform. Theory* 25:495–500, 1979.
30. Kolmogoroff, A. N., and S. V. Fomin, *Elements of the Theory of Functions and Functional Analysis.* New York: Graylock Press, 1957.
31. Korenberg, M. J. Aspects of time-varying and nonlinear systems theory with biological applications. Montréal, Quebec, Canada: McGill University, Ph.D. Thesis, 1972.
32. Korenberg, M. J. Identification of biological cascades of linear and static nonlinear systems. *Proc. 16th Midwest Symp. Circ. Theory* 18.2:1–9, 1973.
33. Korenberg, M. J. Cross-correlation analysis of neural cascades. *Proc. 10th Ann. Rocky Mountain Bioeng. Symp.* 1: 47–52, 1973.
34. Korenberg, M. J. Statistical identification of parallel cascades of linear and nonlinear systems. *IFAC Symp. Ident. Sys. Param. Est.* 1:580–585, 1982.
35. Korenberg, M. J. Statistical identification of Volterra kernels of high order systems. *IEEE Int. Symp. Circ. Sys.* 2:570–575, 1984.
36. Korenberg, M. J. Identifying noisy cascades of linear and static nonlinear systems. *IFAC Symp. Ident. Sys. Param. Est.* 1:421–426, 1985.

37. Korenberg, M. J. Functional expansions, parallel cascades, and nonlinear difference equations. In: *Advanced Methods of Physiological System Modeling,* vol. 1, edited by V. Z. Marmarelis. Los Angeles: Biomedical Simulations Resource, University of Southern California, 1987, pp. 221–240.
38. Korenberg, M. J. Identifying nonlinear difference equation and functional expansion representations: the fast orthogonal algorithm. *Ann. Biomed. Eng.* 16:123–142, 1988.
39. Korenberg, M. J. A fast orthogonal search method for biological time-series analysis and system identification. *Proc. IEEE Int. Conf. Sys. Man. Cybern.* Cambridge, MA: 1989, pp. 459–465.
40. Korenberg, M. J. A robust orthogonal algorithm for system identification and time-series analysis. *Biol. Cybern.* 60: 267–276, 1989.
41. Korenberg, M. J. A rapid and accurate method for estimating the kernels of a nonlinear system with lengthy memory. 15th Biennial Symposium Communication, Queen's University, Kingston, Ontario, Canada, 1990.
42. Korenberg, M. J. Some new approaches to nonlinear system identification and time-series analysis. *Proc. 12th Annu. Int. Conf. IEEE Eng. Med. & Biol. Soc.* 12(1):20–21, 1990.
43. Korenberg, M. J. Parallel cascade identification and kernel estimation for nonlinear systems. *Ann. Biomed. Eng.* 19: 429–455, 1991.
44. Korenberg, M. J., S. B. Bruder, M. Mancini, and P. J. McIlroy. Exact kernel estimation using a noise input: measuring invariant characteristics of cortical cells. Ninth International Conference on Noise in Physical Systems, edited by C. M. Van Vliet. Tea Neck, NJ: World Scientific Publishing Co., 1987, pp. 583–588.
45. Korenberg, M. J., S. B. Bruder, and P. J. McIlroy. Exact orthogonal kernel estimation from finite data records: extending Wiener's identification of nonlinear systems. *Ann. Biomed. Eng.* 16:201–214, 1988.
46. Korenberg, M. J., and I. W. Hunter. The identification of nonlinear biological systems: LNL cascade models. *Biol. Cybern.* 55:125–134, 1986.
47. Korenberg, M. J, and I. W. Hunter. The identification of nonlinear biological systems: Wiener kernel approaches. *Ann. Biomed. Eng.* 18:629–654, 1990.
48. Korenberg, M. J., H. M. Sakai, and K.-I. Naka. Dissection of the neuron network in the catfish inner retina. III. Interpretation of spike kernels. *J. Neurophysiol.* 61:1110–1120, 1989.
49. Krenz, W. C., and L. W. Stark. The interpretation of functional series expansions. *Ann. Biomed. Eng.* 19:485–508, 1991.
50. Lee, Y. W. Contribution of Norbert Wiener to linear theory and nonlinear theory in engineering. In: *Selected Papers of Norbert Wiener,* SIAM, Cambridge, MA: MIT Press, 1964, pp. 17–33.
51. Lee, Y. W., and M. Schetzen. Measurement of the Wiener kernels of a non-linear system by crosscorrelation. *Int. J. Contr.* 2:237–254, 1965.
52. Litt, H. I. A Nonlinear Kernel Investigation of Magnetic Resonance Imaging and Computed Tomography. Buffalo, NY: State University of New York at Buffalo, Ph.D. Thesis, 1994.
53. LYSIS (Ver. 5) Software Program. Los Angeles: Biomedical Simulations Resource, University Southern California.
54. Marmarelis, P. Z., and K. I. Naka. White-noise analysis of a neuron chain: an application of the Wiener theory. *Science* 175:1276–1278, 1972.
55. Marmarelis, P. Z., and K. I. Naka. Nonlinear analysis and synthesis of receptive-field responses in the catfish retina. *J. Neurophys.* 36:605–648, 1973.
56. Marmarelis, P. Z., and V. Z. Marmarelis, *Analysis of Physiological Systems.* New York: Plenum Press, 1978, 487 pp.
57. Marmarelis, V. Z. Identification of nonlinear biological systems using Laguerre expansions of kernels. *Ann. Biomed. Eng.* 21:573–589, 1993.
58. Marmarelis, V. Z., K. H. Chon, Y.-M. Chen, D. J. Marsh, and N.-H. Holstein-Rathlou. Nonlinear analysis of renal autoregulation under broadband forcing conditions. *Ann. Biomed. Eng.* 21:591–603, 1993.
59. Mohler, R. *Nonlinear Systems: Vol. 1—Dynamics and Control and Vol. 2—Applications to Bilinear Control.* Englewood Cliffs, NJ: Prentice-Hall, 1991, 470.
60. Nabet, B., and R. B. Pinter. Multiplicative inhibition and Volterra series expansion. In: *Nonlinear Vision: Determination of Neural Receptive Fields, Function, and Networks,* edited by R. B. Pinter and B. Nabet. Boca Raton, FL: CRC Press, 1992, pp. 475–491.
61. Ogura, H. Estimation of Wiener kernels of a nonlinear system and fast algorithm using digital Laguerre filters. Proceedings of the 15th NIBB Conference on Information Processing in Neuron Network: White-Noise Analysis, edited by K-I. Naka and Y.-I. Ando. Okazaki, Japan: National Institute for Basic Biology, 1986, pp. 14–62.
62. Palm, G., and T. Poggio. Stochastic identification methods for nonlinear systems: an extension of the Weiner theory. *J. Appl. Math.* 34:524–534, 1978.
63. Pece, A. E. C., A. S. French, M. J. Korenberg, and J. E. Kuster. Nonlinear mechanisms for gain adaptation in locust photoreceptors. *Biophys. J.* 57:733–743, 1990.
64. Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, eds. *Numerical Recipes in C.* 2nd edition. New York: Cambridge University Press, 1992, 994 pp.
65. Rice, J. R. A theory of condition. *J. Numer. Anal.* 3:287–310, 1966.
66. Rugh, W. J. *Nonlinear System Theory: The Volterra/Wiener Approach.* Baltimore: Johns Hopkins University Press. 1981, 325 pp.
67. Sakai, H. M., and K.-I. Naka. Signal transmission in the catfish retina. IV. Transmission to ganglion cells. *J. Neurophysiol.* 58:1307–1328, 1987.
68. Sakai, H. M., and K.-I. Naka. Signal transmission in the catfish retina. V. Sensitivity and circuit. *J. Neurophysiol.* 58:1329–1350, 1987.
69. Sakai, H. M., and K.-I. Naka. Dissection of the neuron network in the catfish inner retina. I. Transmission to ganglion cells. *J. Neurophysiol.* 60:1549–1567, 1988.
70. Sakai, H. M., and K.-I. Naka. Dissection of the neuron network in the catfish inner retina. II. Interactions between ganglion cells. *J. Neurophysiol.* 60:1568–1583, 1988.
71. Sakai, H. M., and K.-I. Naka. Dissection of the neuron network in the catfish inner retina. IV. Bidirectional interactions between amacrine and ganglion cells. *J. Neurophysiol.* 63:105–119, 1990.
72. Sakai, H. M, and K.-I. Naka. Dissection of the neuron network in the catfish inner retina. V. Interactions between NA and NB amacrine cells. *J. Neurophysiol.* 63:120–130, 1990.
73. Sakai, H. M., K.-I., Naka, and M. J. Korenberg. White

noise analysis in visual neuroscience. *Vis. Neurosci.* 1: 287–296, 1988.
74. Sakuranaga, M., Y-J. Ando, and K-I. Naka. Dynamics of ganglion cell response in the catfish and frog retinas. *J. Gen. Physiol.* 90:229–259, 1987.
75. Sakuranaga, M., S. Sato, E. Hida, and K-I. Naka. Nonlinear analysis: mathematical theory and biological applications. *CRC Crit. Rev. Biomed. Eng.* 14:127–184, 1986.
76. Sandberg, I. W. Expansions for discrete-time nonlinear systems. *Circuit Sys. Signal Process.* 2:179–192, 1983.
77. Sandberg, A., and L. Stark. Wiener G-function analysis as an approach to nonlinear characteristics of human pupil light reflex. *Brain Res.* 11:194–211, 1968.
78. Schetzen, M. Measurement of the kernels of a nonlinear system of finite order. *Int. J. Control* 1:251–263, 1965.
79. Schetzen, M., *The Volterra and Wiener Theories of Nonlinear Systems* (second edition). New York: John Wiley & Sons, 1980, 531 pp.
80. Schetzen, M. Nonlinear system modeling based on the Wiener theory. *Proc. IEEE* 69:1557–1573, 1981.
81. Shinners, S. M., *Modern Control System Theory and Application*. Boston: Addison-Wesley, 1972, 528 pp.
82. Spekreijse, H., and H. Oosting. Linearizing: a method for analyzing and synthesizing nonlinear systems. *Kybernetik* 7:22–31, 1970.
83. Stark, L. W. Stability, oscillation, and noise in the human pupillary servo mechanism. *Bol. del Inst. de Estudios Medicos y Biologicos* 21:201–222, 1963.
84. Stark, L. W. Neurological control systems: studies in bioengineering, New York: Plenum Press, 1968, 428 pp.
85. Stark, L. W. The pupillary control system: its nonlinear adaptive and stochastic engineering design characteristics. *Automatica* 5:655–676, 1969.
86. Stark, L. W., Y. Takahashi, and G. Zames. Nonlinear servoanalysis of human lens accommodation. *IEEE Trans. Sys. Sci. Cyber.* 1:75–83, 1965.
87. Strang, G., Linear Algebra and its Applications. (second edition) New York: Academic Press, 1980.
88. Sunay, M. O., and M. M. Fahmy. An orthogonal approach to the spatial-domain design of 2-D recursive and nonrecursive nonlinear filters. *IEEE Trans. Circ. Sys.* 41:669–677, 1994.
89. Sutter, E. E. A practical non-stochastic approach to nonlinear time-domain analysis. In: *Advanced Methods of Physiological System Modeling,* vol. I., edited by V. Z. Marmarelis. Los Angeles: Biomedical Simulations Resource, University of Southern California, 1987, pp. 303–315.
90. Sutter, E. E. A deterministic approach to nonlinear systems analysis. In: *Nonlinear Vision: Determination of Neural Receptive Fields, Function, and Networks,* edited by R. B. Pinter and B. Nabet. Boca Raton, FL: CRC Press, 1992, pp. 171–220.
91. Sutter, E. E., and D. Tran. The field topography of ERG components in man. I. The phototopic luminance response. *Vis. Res.* 32:443–446, 1992.
92. Victor, J. D. The fractal dimension of a test signal: implications for system identification procedures. *Biol. Cybern.* 57:421–426, 1987.
93. Victor, J. D., and B. W. Knight. Nonlinear analysis with an arbitrary stimulus ensemble. *Q. Appl. Math.* 37:113–136, 1979.
94. Volterra, V., Leçons sur les Fonctions de Lignes. Paris: Gauthier-Villars, 1913.
95. Volterra, V., *Theory of Functionals and of Integral and Integro-Differential Equations.* New York: Dover, 1959.
96. Watanabe, A., and L. Stark. Kernel method for nonlinear analysis: identification of a biological control system. *Math. Biosci.* 27:99–108, 1975.
97. Wiener, N., *Nonlinear Problems in Random Theory.* Cambridge, MA: MIT Press, 1958.
98. Barahona, M., and C.-S. Poon. Detection of nonlinear dynamics in short, noisy time series. *Nature* 381:215–217, 1996.

Note added in proof: A very recent application of the fast orthogonal algorithm (38) was made by Barahona and Poon (98) for detecting deterministic dynamics in experimental time series.