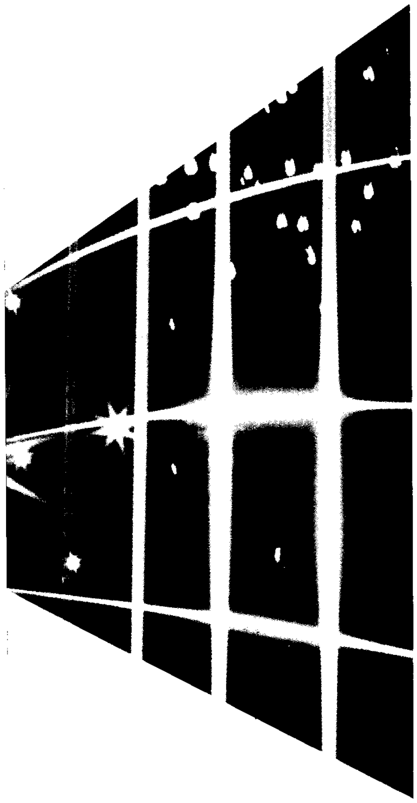
# Orthogonal Approaches to Time-Series Analysis and System Identification

MICHAEL J. KORENBERG and LARRY D. PAARMANN



*Some recent, efficient, orthogonal approaches to system identification and time-series analysis are described and illustrated in this paper. A common thread of most of these approaches is a rapid search for significant terms to include in the model for the system or time series. In the case of system identification, the model generally takes the form of a nonlinear or linear (e.g., ARMA) difference equation where the significant terms, as well as coefficients, are to be estimated. The system may also be represented by a functional expansion (e.g., Volterra series), where the kernels are to be estimated, or by a parallel-cascade arrangement, where the component dynamic linear and static nonlinear elements are to be estimated.*

*In the case of time-series analysis, several model structures may also be adopted. One model structure is a parsimonious sinusoidal series (non-Fourier) where the significant frequencies, amplitudes, and phases are to be estimated in order to approximate concisely the time-series data. Other model structures of interest are autoregressive (AR), moving average (MA), and autoregressive / moving average (ARMA). There is considerable overlap between system identification and time-series analysis; a notable difference is that in time-series analysis, generally there is no access to the model "input". However, we have subsumed under "system identification" the ARMA modeling of time-series y[n] even when it includes an input time-series x[n] which is not measurable.*

*The emphasis of the present paper is on introducing the reader to some useful modeling approaches with sufficient detail and references provided for ready implementation. Also, a tutorial style has been adopted throughout.*

Photo credit: FPG

1053-5888/91/0700-0028$1.00©1991 IEEE

This paper is concerned with some recent methods for systematically constructing models of linear or nonlinear systems, and of time-series data. The need for discrete-time system modeling and modeling of time-series data (discrete-time sample points of a signal or other empirical data) arises in many situations. For example, in order to design a discrete-time control system an accurate discrete-time model of the system to be controlled is required. Moreover, an accurate discrete-time model of a physical system may yield insight into features of the physical system. In addition, an accurate model may be used to predict new data points before they occur. Since these algorithms are very general, applications are very broad in the areas of linear / nonlinear systems identification and signal modeling. Specific applications (for illustration) of these methods include modeling of speech and biological signals such as electrocardiogram (ECG), electromyogram (EMG), and electroencephalogram (EEG); model-based spectral analysis; equalization of channels; cancellation of intersymbol interference; etc.

For system identification, the models in this paper are generally of the following form [1],[2]:

$$\underline{\boldsymbol{y}}[n] = \underline{\boldsymbol{F}}\{\, \underline{\boldsymbol{y}}[n-1],\ldots,\underline{\boldsymbol{y}}[n-K],\underline{\boldsymbol{x}}[n],\ldots,\underline{\boldsymbol{x}}[n-L] \,\} + \underline{\boldsymbol{e}}[n] \quad (1)$$

where $\underline{\boldsymbol{x}}[n]$, $\underline{\boldsymbol{y}}[n]$ are respectively the system multidimensional input and output, $\underline{\boldsymbol{F}}$ is the multidimensional system function, and $\underline{\boldsymbol{e}}[n]$ is the multidimensional model error. When $\underline{\boldsymbol{F}}$ is nonlinear, (1) can be a nonlinear difference equation. Equation (1) includes autoregressive (AR), moving average (MA), and autoregressive / moving average (ARMA) model representations ($\underline{\boldsymbol{F}}$ linear): examples of linear systems. Equation (1) also includes the finite-memory, finite-order discrete-time Volterra series ($\underline{\boldsymbol{F}}$ a polynomial function of $\underline{x}[n],\ldots,\underline{x}[n-L]$ only): an example of a nonlinear system. It is assumed throughout this paper that the system is a continuous functional or mapping of the input (in that small changes in the system input result in small changes in the system output, i.e., discontinuities are excluded), time-invariant, causal, and depends to an arbitrarily small extent on the remote past of the input. Systems with hysteresis are excluded. For simplicity, throughout the remainder of this paper, only single-input, single-output discrete-time systems will be considered.

For time-series analysis, the model is generally of the following form:

$$y[n] = \sum_{m=0}^{M} a_m p_m[n] + e[n] \qquad (2)$$

Here, the $p_m[n]$ terms may be past values of y[n] (AR model), present and past values of x[n] (MA model), a combination (ARMA model), or a wide variety of other functions such as triangular waveforms, square waves, exponentials or exponentially-decaying sinusoids, logarithmic functions, splines, polynomials, or sine and cosine functions where the frequencies (not necessarily commensurate) are to be estimated. However, since ARMA models are contained within (1), they are considered below under "system identification" to streamline the presentation, and sinusoidal series will be presented below under "time-series analysis".

For both system identification and time-series analysis, a common problem (and a principal preoccupation of this paper) is the selection of terms to include in the model, so that a concise representation is constructed for the system or time-series data. For example, suppose a difference equation model is to be constructed for a nonlinear system. Then the multidimensional $\underline{\boldsymbol{F}}$ (polynomial considered here) may comprise a constant, a linear combination of x and y terms on the right side of (1), and crossproducts (including powers) of any number of these terms. As several authors have pointed out, e.g. [3], it is important to select a concise subset out of all possible candidate terms in order to build an economical (parsimonious) but accurate model of the nonlinear or linear system.

Similarly, suppose that given time-series data are to be accurately approximated by a linear combination of sine and cosine functions. A Fourier series representation could be used, wherein the frequencies of the sines and cosines are harmonics (integer multiples) of the fundamental frequency corresponding to the record length. However, this may result in an unnecessarily long sinusoidal series because the significant frequencies in the given time-series may be unrelated to the particular record length available. A much more economical sinusoidal series representation can be systematically developed for the time-series by orthogonally searching [4],[5] for the most significant sinusoidal frequencies first (see below). Moreover, in the methods examined below, the frequencies selected for the sinusoidal series representation are not required to be commensurate nor integer multiples of a fundamental frequency. This is an important advantage which can achieve parsimony of the sinusoidal series representation, and much finer frequency resolution (up to eight times better, as shown below) compared to the conventional Fourier series approach.

The examination below is divided into sections on system identification and time-series analysis, but there is some overlap between the sections. For example, the fitting of ARMA models to time-series data is frequently considered to be system identification with an inaccessible input, and this is the classification followed here.

## SYSTEM IDENTIFICATION

This section examines some recent methods for systematically constructing difference equation models of linear [6] or nonlinear [4],[5],[7],[8] systems of unknown structure, and for estimating the kernels in a functional expansion of a nonlinear system [9],[10], particularly by building parallel-cascade representations of nonlinear systems [11]-[14]. A common theme in most of the presentation is the use of orthogonal approaches to simplify the model development. Wiener [15] pioneered the use of the Gram-Schmidt procedure to construct a functional expansion where the terms are orthogonal for a given white Gaussian input process. The Wiener kernels in his orthogonal series can be estimated in the time domain via the crosscorrelation formula of Lee and Schetzen [16], or in the frequency domain via the method of French and Butz [17]. Important extensions [18] and applications [19]-[21] of the Wiener-Lee-Schetzen approach have followed. This work has indeed motivated several of the methods examined below.

# DIFFERENCE EQUATION MODELING

The key issue considered here is the efficient selection of difference equation terms to build a concise, accurate model of a nonlinear system. The term-selection techniques utilize well-known classical or modified Gram-Schmidt orthogonalization and Cholesky decomposition. The recent algorithmic developments which are focused upon below concern how Gram-Schmidt and Cholesky procedures can be used as part of a method for searching out significant model terms.

An orthogonal search method was developed [7],[8] for systematically constructing difference equation models of nonlinear (which includes the linear case) dynamic systems with unknown structure. A very similar approach had previously been used by Desrochers [22] to build static models of nonlinear, zero-memory (i.e., nondynamic) systems.

## Orthogonal Search Method

Suppose that the input x[n] and the output y[n] are available for some system for $n$=0,...,$N$. (The experiment to obtain the data may have begun well before, and continued long after, the available record segment.) The desired difference equation model will have the form of (2), where

$$p_0[n] = 1 \tag{3}$$

and for $m \geq 1$,

$$p_m[n] = y[n-k_1]\ldots y[n-k_i]x[n-l_1]\ldots x[n-l_j] \tag{4}$$

where

$$i \geq 0, \quad 1 \leq k_1 \leq K, \ldots, 1 \leq k_i \leq K \tag{5a}$$

$$j \geq 0, \quad 0 \leq l_1 \leq L, \ldots, 0 \leq l_j \leq L \tag{5b}$$

In (5), $i$= 0 denotes that $p_m[n]$ does not contain any y terms. Similarly, $j$= 0 denotes that $p_m[n]$ does not contain any x terms.

Let $N_0 = \max(K, L)$ be the maximum input or output lag permitted in the candidate difference equation terms of (4), where $N_0<N$. Then, each $p_m[n]$ will have the form shown in (3) or (4), and the data will be complete (no argument negative for any x or y term) for $n = N_0,\ldots,N$. According to the orthogonal search method, the difference equation represented by (2) may be rewritten as the following orthogonal series:

$$y[n] = \sum_{m=0}^{M} g_m w_m[n] + e\,[n]$$

In (6), the $w_m[n]$ are constructed from the $p_m[n]$ (via the Gram-Schmidt or, preferably, the modified Gram-Schmidt procedure) to be mutually orthogonal over the interval $n = N_0,\ldots,N$. The $g_m$ are selected to minimize the mean-square error (mse) over this interval, namely

$$\overline{e^2[n]} \triangleq \frac{1}{N-N_0+1}\sum_{n=N_0}^{N} e^2[n] = \overline{\left(y\,[n] - \sum_{m=0}^{M} g_m w_m[n]\right)^2}$$

$$= \overline{y^2[n]} - \sum_{m=0}^{M} \overline{g_m^2\, w_m^2[n]} \tag{7}$$

In (7) and elsewhere, the overbar denotes the time-average from $n = N_0$ to $n = N$. The fact that the mean-square error is minimized over this truncated interval ensures that no error is introduced by the unavailability of the data prior to $n$ = 0 (i.e., for negative arguments of x and y).

Suppose that $a_r p_r[n]$ was the last difference equation term added to the model in (2). Then it is easily shown from (7) that the addition of this term reduced the mean-square error by the following amount [4]-[8]:

$$Q[r] = g_r^2\overline{w_r^2[n]} \tag{8}$$

where

$$g_r = \frac{\overline{y[n]w_r[n]}}{\overline{w_r^2[n]}} \tag{9}$$

Equation (8) follows from the fact that the $w_r[n]$ were constructed to be mutually orthogonal. Consider in more detail how the $a_r p_r[n]$ term can be selected. Thus, to expand the model by a further difference equation term, evaluate the quantity Q in (8) for each candidate addition to the model. Choose the candidate for which Q is greatest, since the addition of this term will result in the largest reduction in mean-square error. As noted above, Desrochers [22] has previously used this approach (and an analogous equation to (8)) to fit static nonlinear models. By continuing to select terms in this manner, one may construct a concise but accurate difference equation model for a real system, particularly if a threshold is used to reject unsuitable terms [8].

Note however that the construction of the orthogonal functions $w_m[n]$, especially since this must be done for all candidate terms, is computationally intensive, and also can require excessive amounts of computer memory. The Fast Orthogonal Search, to be described next, avoids these problems.

## Fast Orthogonal Search

Fast Orthogonal Search [4],[5] was developed as a more efficient method for building difference equation models of nonlinear systems, and for modeling time-series data. Suppose first that all $p_m[n]$ terms in (2) were known a priori. Then, to estimate the $a_m$ use a Cholesky decomposition, which can be carried out using the following pseudo-code (for similar substitute code, e.g. the Cholesky outer product version, see Golub and Van Loan [23]):

```
$D[0, 0] = 1$                                            (10)
 FOR $m = 1$ TO $M$
$D[m, 0] = \overline{p_m[n]}$                            (11)
  FOR $r = 1$ TO $m$
$P[m, r] = \overline{p_m[n]p_r[n]}$                      (12)
  NEXT $r$
  NEXT $m$
  FOR $m = 1$ TO $M$
  FOR $r = 0$ TO $m-1$
$A[m, r] = D[m, r] \,/\, D[r, r]$
  $D[m, r+1] = P[m, r+1] - \sum_{i=0}^{r} A[r+1, i]\, D[m, i]$
  NEXT $r$
  NEXT $m$
$C[0] = \overline{y[n]}$                                 (13)
  FOR $m = 1$ TO $M$
$C[m] = \overline{y[n]p_m[n]} - \sum_{r=0}^{m-1} A[m, r]\, C[r]$    (14)
  NEXT $m$
  FOR $m = 0$ TO $M$
$G[m] = C[m] \,/\, D[m, m]$                              (15)
  NEXT $m$
```

(The above pseudocode is merely for illustration and clarity of presentation; modification to the code can readily be made to improve efficiency, but that is not the intent here.)

Finally, the coefficients $a_m$ in the difference equation represented by (2) can be calculated by the following [9]:

$$a_m = \sum_{i=m}^{M} G[i]\, v[i] \tag{16}$$

where

$$v[m] = 1 \tag{17}$$

$$v[i] = -\sum_{r=m}^{i-1} A[i, r]\, v[r], \qquad i = m+1, \dots, M \tag{18}$$

However, in practice the particular choices of the $p_m[n]$ for achieving an economical model are not usually known in advance. Therefore, a simple term selection procedure is now discussed, but it will be appreciated that many variations of this procedure can be readily set down. First, use (3) to introduce a constant term into the difference equation model, and note that (in view of (10) and (15)) G[0] is given by (13). Then, with M successively set equal to 1, 2, ... , each $p_M[n]$ model term may be chosen as follows. For each candidate for $p_M[n]$, denoted $\underline{p}_M[n]$, evaluate the quantity [4],[5]

$$\underline{Q}[M] = \underline{G}^2[M]\, \underline{D}[M, M] \tag{19}$$

where here, and throughout the remainder of the paper, an underscore indicates a candidate function or its coefficient. It can be shown that $D[M,M]$ equals $\overline{w_M^2[n]}$, and indeed equation (19) is equivalent to (8) and thus follows from the mutual orthogonality of the $w_m[n]$. Equation (19) provides the reduction in mse if one further term, namely $\underline{p}_M[n]$, were to be added to the model. Therefore, choose the candidate for which $\underline{Q}$ in (19) is the greatest (optionally, subject to exceeding a specified threshold level), since this choice for $p_M[n]$ will cause the greatest reduction in mse.

To calculate $\underline{G}[M]$ and $\underline{D}[M,M]$ in (19) for a given candidate $\underline{p}_M[n]$, one may shorten the above pseudocode (from (10) to the line following (15)) by carrying it out only for $m=M$, and not for earlier values of m. This avoids repeating calculations already performed at earlier rounds of searching. After selecting $p_M[n]$, repeat the abbreviated pseudocode for the chosen candidate, to properly set the values of G[M], D[M,M], etc., before continuing the process. To use the abbreviated pseudocode requires knowing the time-averages (which are always from $n=N_0$ to $n=N$) on the right sides of (11), (12), (13), and (14). The time-average in (13) can be computed once at the outset and stored. The time-averages in (11), (14), and (12) when $r=m$, can also be calculated initially (letting $p_m[n]$ equal in turn each candidate term) and stored. Then, to be able to use the abbreviated pseudocode to calculate $\underline{G}[M]$ and $\underline{D}[M,M]$ in (19) for a given candidate, one need merely determine the time-average of that candidate with the last-selected term. All other required time-averages will be available from the previous rounds of searching.

Moreover, the lagged nature of the difference equation terms in (4) makes it possible to accelerate the calculation of the time-averages on the right sides of (11) and (12). This can be done by relating the time-averages to input and output means and correlations, and then making small corrections for the finite record length. Such an approach is much more efficient than computing each time-average independently, and moreover avoids the need to explicitly create and store the $p_m[n]$ terms. To illustrate, suppose that

$$p_m[n] = y\,[n-\alpha], \quad \alpha \geq 1 \tag{20}$$

Then, the time-average of (20) (required on the right side of (11)) would be

$$\overline{p_m[n]} = \overline{y[n]} + \frac{1}{N-N_0+1}\sum_{i=1}^{\alpha} (y\,[N_0-i] - y\,[N-i+1]), \tag{21}$$

where

$$\overline{y\,[n]} = \frac{1}{N-N_0+1}\sum_{n=N_0}^{N} y\,[n]$$

As noted, the output average is computed at the outset. Then the time-average of any $p_m[n]$ of the form of (20) can be calculated from (21) to make the corrections for the finite record length. (These corrections could also be made recursively [14],[26].)

Similarly, suppose that

$$p_m[n] = y\,[n-\alpha]\; x\,[n-\beta], \quad \alpha \geq 1, \quad \beta \geq 0 \tag{22}$$

It is assumed that for $r \geq 0$

$$\varphi_{xy}[r] = \frac{1}{N-N_0+1}\sum_{n=N_0}^{N} x\,[n]\; y\,[n-r], \tag{23}$$

and

$$\varphi_{yx}[r] = \frac{1}{N-N_0+1}\sum_{n=N_0}^{N} y\,[n]\; x\,[n-r]$$

have been computed at the outset. If $\beta = 0$ in (22), then the time-average on the right side of (11) is given by (23) with $r = \alpha$. Otherwise,

$$\overline{p_m[n]} = \varphi_{xy}[\alpha-\beta] + \frac{1}{N-N_0+1}\sum_{i=1}^{\beta}(x[N_0-i]y[N_0-i+\beta-\alpha]-x[N-i+1]y[N-i+1+\beta-\alpha]) \qquad 1 \leq \beta \leq \alpha,$$

and

$$\overline{p_m[n]} = \varphi_{yx}[\beta-\alpha] + \frac{1}{N-N_0+1}\sum_{i=1}^{\alpha}(y[N_0-i]x[N_0-i+\alpha-\beta]-y[N-i+1]x[N-i+1+\alpha-\beta]) \qquad 1 \leq \alpha \leq \beta$$

so that the required time-averages can be efficiently calculated simply by making small changes in $\varphi_{xy}$ or $\varphi_{yx}$. The time-averages on the right side of (12) can be calculated analogously from the input and output correlations. See references [14],[26] for illustration of how required time-averages can be efficiently calculated recursively.

It is stressed that there is no need to create or store the candidate difference equation terms (which may well number in the thousands). Instead, only the averages of these terms, and averages of certain pairs of these terms, are required and can be efficiently obtained as illustrated above.

For the difference equation model represented by (2), the mse is (from (7), (8) and (19))

$$\overline{e^2[n]} = \overline{y^2[n]} - \sum_{m=0}^{M} G^2[m]\; D\,[m,m] \tag{24}$$

and this may be used in deciding when to stop adding further terms. Model development may also be halted when a predetermined number of terms have been selected, or when no remaining candidate can cause a reduction in mse exceeding a specified threshold level. A simple statistical criterion for rejecting unsuitable candidates is set out below.

## Discussion of Search Techniques

It will be clear to the reader that difference equation development via the orthogonal search method and via fast orthogonal search are closely related, as typified by the equivalence of (8) and (19). Fast orthogonal search is much faster and requires $O(MN_1P+M^3P)$ multiplications in total, where P is the number of candidate terms, M << P is the number of (nonconstant) terms in the final model, and $N_1 = N-N_0+1$. This number of computations is needed if each time-average required in fast orthogonal search were computed independently. In fact, the number of computations is reduced since, as illustrated above, important savings in calculating time-averages are available by exploiting the lagged nature of the difference equation terms. Both the orthogonal search method and fast orthogonal search, either explicitly or implicitly, orthogonalize the candidate terms, and then use (8) or (19) to test each candidate and choose the one resulting in the greatest reduction of mse. Desrochers [22] has previously used this approach with an equivalent equation to build static models of nonlinear, zero-memory systems. Some of the scalar equations in fast orthogonal search are similar to Desrochers' matrix equations, but while his work is extremely valuable, there are significant differences from fast orthogonal search (see [24]).

First, Desrochers' method requires $O(N_1P^{2)}$ multiplications just to construct an initial matrix, where $N_1$ is the length of his input sequence. Second, Desrochers' work did not consider modeling by difference equations, and so could not take advantage of computational savings inherent in computing time averages involving difference equation terms as illustrated above. Third, Desrochers' approach determines a concise static model, but does not yield the model coefficients. Desrochers and Mohseni [25] propose a related approach which they apply to modeling dynamic nonlinear systems via differential equations. Again, the work does not exploit the computational savings inherent in a difference equation approach.

The orthogonal search method [7],[8] for fitting difference equation models can be applied in many variations. For example, to increase speed of searching, McIlroy [8] arranged candidate difference equation terms into disjoint subsets (which were searched successively): (1) linear x terms, (2) linear y terms, (3) xx terms, (4) yy terms, and (5) xy terms. Within a given subset, a candidate term was permanently dropped from consideration if its reduction of mse was less than a specified percentage of the output mean-square. The use of this threshold to permanently discard terms greatly accelerated searching. Once a subset was exhausted, the next subset was searched. This procedure was used to model several real systems, including a data transmission channel [8]. A similar compartmentalized search can optionally be implemented when fast orthogonal search is used; this has been utilized to model dispersive nonlinear data transmission channels [26].

Finally, a simple statistical criterion (a standard correlation test) can optionally be used in deciding whether to reject a candidate term. Suppose model terms up to and including $a_M p_M[n]$ have been selected, so that e[n] is the residual in the model of (2). Suppose that, for a given candidate for $a_{M+1}p_{M+1}[n]$, the corresponding value of Q is calculated, with M replaced by M+1 in (19). It can be shown [27] that if e[n] is zero-mean, independent Gaussian noise, then

$$r = \left[ \frac{G^2[M+1]\, D\,[M+1, M+1]}{\overline{y^2[n]} - \sum_{m=0}^{M} G^2[m]\, D\,[m,m]} \right]^{1/2} < \frac{2}{\sqrt{N-N_0+1}}$$

with probability of about 0.95, for sufficiently long record length $(N-N_0+1)$. Hence, before choosing a candidate for $a_{M+1}p_{M+1}[n]$, one may optionally require that

$$G^2[M+1]D[M+1,M+1] > \frac{4}{N-N_0+1}\left( \overline{y^2[n]} - \sum_{m=0}^{M} G^2[m]D[m,m] \right) \tag{25}$$

For smaller record lengths, replace the factor $4/(N-N_0+1)$ in (25) by the square of the critical correlation coefficient value for sample size $(N-N_0+1)$. [Clearly, a factor corresponding to other than 95% confidence limits may be substituted for the factor in (25).] If there are no candidate terms remaining which satisfy (25), then model development may be halted. In addition, a "single-pass" search can be implemented whereby each candidate satisfying (25) is added to the model, while a candidate which fails this screening is permanently rejected.

## Fitting of ARMA Models

To facilitate understanding of the following discussion, it is useful to describe very briefly the classical Gram-Schmidt and the modified Gram-Schmidt procedures. Suppose again that $p_m[n]$, $m = 0,1,\ldots,M$, are defined over the time-interval $n = N_0,\ldots,N$. Then a set of data files $w_m[n]$, $m = 0,1,\ldots,M$, which are mutually orthogonal over this time-interval can be created via (classical) Gram-Schmidt orthogonalization as follows. Set

$$w_0[n] = p_0[n]$$

and, for $m = 1,2,\ldots,M$,

$$w_m[n] = p_m[n] - \sum_{r=0}^{m-1} \alpha_{mr}\, w_r[n]$$

where

$$\alpha_{mr} = \frac{\overline{p_m[n]\, w_r[n]}}{\overline{w_r^2[n]}}$$

Alternatively, to create a set of data files $p_m^{(m)}[n]$ which are mutually orthogonal over the interval $n = N_0,\ldots,N$, the more robust modified Gram-Schmidt procedure [28] can be used. Define

$$p_m^{(0)}[n] = p_m[n], \quad m = 0,1,\ldots,M$$

and, for $j = 0,1,\ldots,M-1$; $m = j+1,\ldots,M$

$$p_m^{(j+1)}[n] = p_m^{(j)}[n] - \alpha_{mj}\, p_j^{(j)}[n]$$

where

$$\alpha_{mj} = \frac{\overline{p_m^{(j)}[n] p_j^{(j)}[n]}}{\overline{(p_j^{(j)}[n])^2}}$$

Since the orthogonal functions $p_m^{(m)}[n]$ take the place of the functions $w_m[n]$ in the classical Gram-Schmidt procedure, it is understood that the former functions may be substituted for the latter, in (8) and (9) for example. Indeed,

$$g_m = \frac{\overline{y[n]\, p_m^{(m)}[n]}}{\overline{(p_m^{(m)}[n])^2}}, \quad m = 0,\ldots,M$$

The input / output data are to be fit by a linear difference equation, an ARMA model of the following form:

$$y\,[n] = -\sum_{k=1}^{K} c_k\, y\,[n-k] + \sum_{i=0}^{L} b_i\, x\,[n-i] + e\,[n] \tag{26}$$

In (26), K and L again represent the maximum output and input lags permitted for the difference equation terms. However, the identified model may well have a lower order, say $(K_1,L_1)$, where $K_1 \leq K$ and $L_1 \leq L$, with $c_k = 0$, $k{>}K_1$, and $b_i = 0$, $i{>}L_1$.

Since the ARMA model in (26) does not contain a constant term, it is not necessary to define $p_0[n]$ using (3). Instead, $p_0[n]$ can be the first of the terms $x[n-i]$, $i \geq 0$, and $y[n-k]$, $k \geq 1$, chosen for the model. (Alternately, one could of course use (3) to introduce a constant into the model, in which case $p_1[n]$ would represent the first chosen x or y term.) In the ARMA identifier described by Korenberg and Paarmann [6], a candidate x term is paired with a candidate y term; one member of the pair is chosen for the model. The remaining term is paired with a new candidate term whose lag is one greater than the chosen term. The selection process continues until the mse of the model is less than a specified threshold value.

More specifically, the first candidate pair is x[n] and $y[n-1]$. If x[n] is selected for the model, then the next candidate pair is $x[n-1]$ and $y[n-1]$, otherwise the new pair is x[n] and $y[n-2]$. Consider choosing the M-th model term, $p_M[n]$, $M \geq 0$ . For each of the two candidate terms, use the classical (or, preferably, the modified) Gram-Schmidt procedure to create the corresponding $w_M[n]$ (or $p_M^{(M)}[n]$) orthogonal to the model terms *already selected*. Hence, determine from (8) and (9) the mse reduction Q[M], if the candidate term were the sole addition to the model. Choose the member of the pair with the larger Q value. Stop the process when a preset number of model terms have been selected, or when the mse is less than a specified percentage of the output variance (or, the output mean-square). Term selection may also be halted when neither member of the candidate pair can cause a mse reduction greater than a threshold level. Then, (16)-(18) can be used to calculate the difference equation coefficients $a_m$ in (2) and thus $c_k$ and $b_i$ in (26). It will be appreciated that the above identifier can also be used without the searching feature to fit data by ARMA, AR, or MA models of specified order.

Consider the case when the system output y[n] is corrupted by additive zero-mean, stationary, white or colored noise, independent of the system input, and only the noisy output $z[n]$ ( $z[n] = y[n] + v[n]$, where v[n] is the additive noise) and $x[n]$ are measurable. (For a linear system, corruption of the input by such noise is equivalent to contamination of the output by additive zero-mean colored stationary noise independent of the input.) Much of the noise corrupting the output can be removed as follows [6],[12]. Using the noisy output in place of (unmeasurable) $y[n]$, estimate a high-order MA model of the form:

$$z[n] = \sum_{i=0}^{L} b_i x[n-i] + e[n]$$

Use the identified MA coefficients $\hat{b}_i$ to calculate the estimated noise-free output:

$$\hat{y}[n] = \sum_{i=0}^{L} \hat{b}_i x[n-i]$$

For sufficiently long record length, $\hat{y}[n]$ will be significantly closer [6] to the noise-free output y[n] than is the noisy output z[n]. Finally, use $\hat{y}[n]$ as a substitute for y[n] in the above-described ARMA identifier.

A similar modification can be used to fit ARMA models to time-series data y[n], when the input x[n] is inaccessible [6]. This modification follows the approach of Konvalinka and Matausek [29] except that the ARMA identifier proper is different, slightly better results are obtained here [6], and the present method includes automatic order estimation. First, fit the time-series data by a high-order AR model of form:

$$y[n] = -\sum_{k=1}^{K} c_k y[n-k] + e[n] \tag{27}$$

Next, use the identified AR coefficients $\hat{c}_k$ to obtain an estimate of the input:

$$\hat{x}[n] = y[n] + \sum_{k=1}^{K} \hat{c}_k y[n-k] \tag{28}$$

The $\hat{x}[n]$, y[n] data are then utilized in the ARMA identifier with automatic order estimation described above. To iterate the process, use the identified MA coefficients to inverse-filter y[n] [29]. The resulting signal $y_1[n]$ is then used in place of y[n] to fit the high-order AR model of the form in (27). The idea is to remove the MA part for a better high-order AR fit. Next, the newly identified AR coefficients $\hat{c}_k$ are employed to provide an improved estimate of the input (analogous to (28) with y replaced by $y_1$). Then y[n] and the improved estimate of x[n] are used in the above-described ARMA identifier, etc.

*Example 1*

In Example 1 an ARMA system was excited by zero-mean, white, Gaussian noise (x[n]), and the resultant system output, y[n], was corrupted by additive noise yielding z[n]. The additive noise was zero-mean, white, Gaussian noise uncorrelated with the system input. The data lengths for the input and output data were 2000 points each. The AR parameters are -2.41, 2.96, -2.02 and 0.73, and the MA parameters are 1.0, -0.17, 0.29, -0.23 and 0.58.

In the first part of this example both x[n] and y[n] (noise-free data) were made available to the ARMA identification algorithm. The true system parameters were identified essentially without error (identical to ten significant places). Note that the identification algorithm not only accurately identified the system parameters, but correctly estimated the system order as well.

In the second part of this example only x[n] and z[n] were made available to the identification algorithm. The signal-to-noise ratio (SNR) was 10 dB. The MA method of noise reduction described above was first used, with an MA order of 40, yielding $\hat{y}[n]$. The resultant SNR of $\hat{y}[n]$ was 18 dB. This was followed by the ARMA iden-

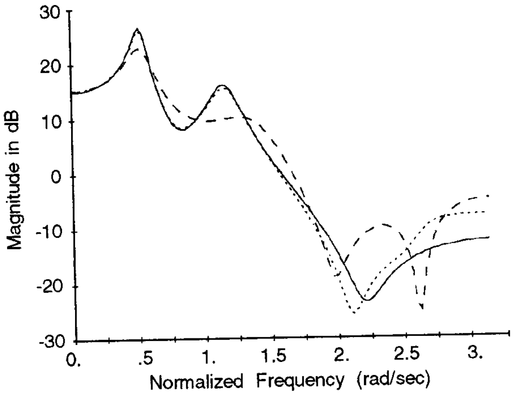


*Fig. 1. Frequency spectra for Example 1. The SNR was 10 dB. Solid line, true; short-dashed line, ARMA estimate that includes MA noise reduction; long-dashed line, ARMA estimate without MA noise reduction.*

tification algorithm; the algorithm input data being x[n] and $\hat{y}[n]$. For comparison, the identification was also conducted without first using the MA method of noise reduction. Also, for comparison, both ARMA identifications used a fixed order of 6,6 (maximum lag for both input and output was 6). For good performance in additive noise, the identification order should exceed the true order; see reference [6] for a discussion.

Figure 1 illustrates the errors in the frequency domain. The true frequency response is shown by the solid line, the frequency response based on the estimated parameters obtained with MA noise reduction is shown by the short-dashed line, and the frequency response based on the estimated parameters obtained without MA noise reduction is shown by the long-dashed line. Note that the short-dashed response matches the true response much better than does the long-dashed response. If plotted, the estimated frequency response for the noise-free case would be superimposed on, and indistinguishable from, the true response.

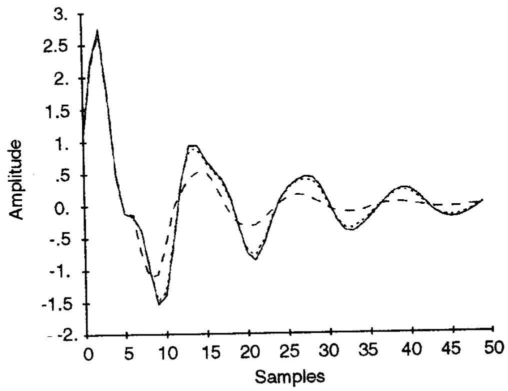


*Fig. 2. Unit sample responses for Example 1. The SNR was 10 dB. Solid line, true; short-dashed line, ARMA estimate that includes MA noise reduction; long-dashed line, ARMA estimate without MA noise reduction.*

Figure 2 illustrates the errors in the time domain; a comparison of unit sample responses. The true response is shown by the solid line, the response based on the estimated parameters obtained with MA noise reduction is shown by the short-dashed line, and the response based on the estimated parameters obtained without MA noise reduction is shown by the long-dashed line. Note that the short-dashed response matches the true response much better than does the long-dashed response. If plotted, the estimated unit sample response for the noise-free case would be superimposed on, and indistinguishable from, the true response.

*Example 2*

In Example 2 an ARMA system was excited by zero-mean, white, Gaussian noise with the following poles:

$0.95e^{\pm j0.2}$, $0.98e^{\pm j1.2}$, $0.90e^{\pm j2.0}$

and with the following zeros:

$1.0e^{\pm j0.70}$, $1.0e^{\pm j2.5}$

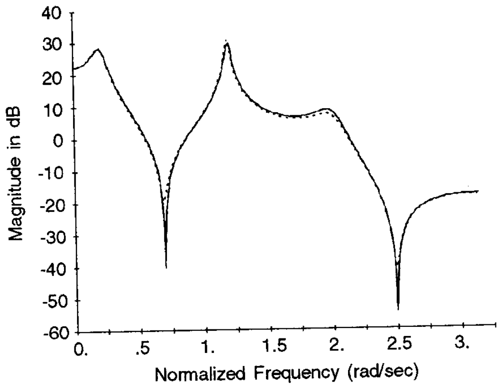


*Fig. 3. Frequency spectra for Example 2. The input was not accessible. Solid line, true; dashed line, based on the ARMA estimate.*

Note that the zeros are on the unit circle. The magnitude frequency spectrum is shown in Fig. 3 as the solid line. Two thousand points of this signal (the output of the ARMA filter) was the only data supplied to the ARMA identification algorithm (the input was not accessible). The algorithm for the case where the input is not accessible was briefly presented above. Details may be found in [6]. The algorithm, not given the ARMA order a priori, estimated six poles (six actual) and four zeros (four actual). The accuracy of the estimated parameters is illustrated in the frequency domain in Fig. 3. The estimated frequency spectrum of the signal is shown as the dashed line, and shows little error relative to the true spectrum.

For an indication of the superior accuracy of the present algorithm applied to ARMA signals (input not accessible), see [6] for a comparison with the method of Konvalinka and Matausek [29]; see [30] for a comparison with six other methods.

# IDENTIFICATION VIA FUNCTIONAL EXPANSIONS

There are various ways of mathematically representing nonlinear systems. A popular representation has been by means of Volterra series. This section considers approximating a nonlinear system both by a discrete-time Volterra series, and by parallel cascades of dynamic linear and static nonlinear elements.

## Volterra Series

As noted in the introduction, the right side of the model in (1) contains the finite-order, finite-memory, discrete-time Volterra series:

$$y_s[n] = \sum_{l=0}^{L} \sum_{i_1=0}^{R} \cdots \sum_{i_l=0}^{R} k_l\,[i_1,\ldots,i_l]\; x\,[n-i_1\,]\ldots x\,[n-i_l\,] \qquad (29)$$

In (29), L is the order (or degree) of the Volterra series, (R+1) is the memory length, and $k_l$ is the (symmetric) $l$-th order Volterra kernel. The zero-order kernel $k_0$ is a constant. For the continuous-time case, it is known [31] that any time-invariant, finite-memory system which is a continuous functional of its input can be uniformly approximated over a uniformly bounded, equicontinuous set of input signals by a Volterra series of sufficient, but finite, order. Consider next a discrete-time system which also has the property that "small" changes in the system input result in "small" changes in the system output, and which furthermore is time-invariant, causal, and possesses finite memory. It follows from the Stone-Weierstrass theorem [32],[33] that such a system can be uniformly approximated over a uniformly bounded set of input signals by a series of the form in (29), for sufficiently large L. The kernels $k_l$ characterize the system in that they enable prediction of the system output corresponding to any member from the set of input signals.

Since finite-order Volterra series are included in the model of (1), kernel estimation [9],[10],[14],[34] can be carried out using the difference equation modeling described above. For example, the pseudocode from (10) to the line following (15) can be used in conjunction with efficient procedures [10],[14],[34] for calculating the time-averages on the right side of (11) and (12) from the input mean and autocorrelations. (Note that in this application $i = 0$ in (4), so that the $p_m[n]$ defined by (3) and (4) contain no y terms.) Then (16)-(18) yield the $a_m$ which are directly related to the desired kernels. For further details, see [9],[10],[14],[34].

However, a very effective method for estimating the kernels of a nonlinear system with lengthy memory relies on a different approach, parallel cascade identification, which is briefly reviewed next.

## Parallel Cascade Identification

Consider again a causal, finite-memory, time-invariant, discrete-time system which is a continuous mapping of its input in that "small" changes in the system input result in "small" changes in the system output. Palm [33] has shown that any such system can be uniformly approximated by a finite sum of cascades, each comprising a dynamic linear, a static nonlinear, and a dynamic linear system. Korenberg [11]-[14] has proposed a method for identifying parallel cascade representations of this form. The representation is related to an approach of Wiener [15] and Bose [35] for expanding Wiener kernels using a complete set of basis functions (e.g., the Laguerre functions). However, a fixed set of basis functions may not result in rapid convergence to accurately represent a wide variety of nonlinear systems. In the parallel cascade method [11]-[14], the first component in each path, a dynamic linear system, is defined using a slice of a crosscorrelation function, so that, in effect, the basis functions change with the system to be identified.

For the parallel cascade method to apply to any nonlinear, causal, finite-memory, discrete-time, time-invariant system which is a continuous mapping of the input, it suffices if each of the parallel paths comprises a dynamic linear system followed by a static nonlinearity [11]. The model may be represented as follows:

$$y\,[n] = \sum_{i=1}^{I} z_i[n] + e\,[n]$$

where $z_i[n]$ is the output of the i-th linear/nonlinear cascade path, I is the number of parallel cascade paths, and e[n] is the final residue error. For simplicity, identification via such simple parallel cascades will be described here, although one can continue to add dynamic linear systems and static nonlinearities in developing each path [11]-[14].

The essence of parallel cascade identification is to approximate the nonlinear system output using a first cascade, compute the residue between system and cascade outputs, then approximate the residue using a second cascade, etc. For a very broad class of systems the sum of the cascades can provide an arbitrarily close approximation, in the mean-square sense, to the system to be modeled. (For proof of convergence of the algorithm, see reference [14].) The discrete impulse response of the first component in each path is defined via a slice of a crosscorrelation (first or higher order) of the input with the residue remaining after estimating the previous cascade. Suppose that $y_i[n]$, $n = 0,\ldots,N$ is the residue remaining ($y[n] - \sum_{j=1}^{i} z_j[n]$) after estimating the i-th cascade (where $y_0[n] = y[n]$). Consider approximating the residue $y_{i-1}[n]$ by an i-th cascade, $i \geq 1$.

Let $h_i[n]$ be the (discrete) impulse response of the dynamic linear system in the i-th cascade path, and $z_i[n]$ be the cascade output. Then,

$$y_i[n] = y_{i-1}[n] - z_i[n] \quad (30)$$

Let $h_i[m]$ be randomly set equal to one of

$$\varphi_{xy_{i-1}}[m] = \overline{y_{i-1}[n]\, x[n-m]} \quad (31)$$

and

$$\varphi_{xxy_{i-1}}[m,A] \pm E\, \delta[m-A] \quad (32)$$

where $m = 0,\ldots,R$, and the overbar denotes the time-average over the interval from $n = R$ to $n = N$. The *slice* of the second-order crosscorrelation $\varphi_{xxy_{i-1}}$ in (32) is likewise computed by time-averaging over this interval. Moreover, in (32) the discrete delta function $\delta[n] = 0$, $n \neq 0$, and $\delta[0] = 1$, the sign of the delta term is chosen at random, the constant A is randomly selected from $0,\ldots,R$, and the parameter E is made to tend to zero as the mean-square of the residue approaches zero. For example, set

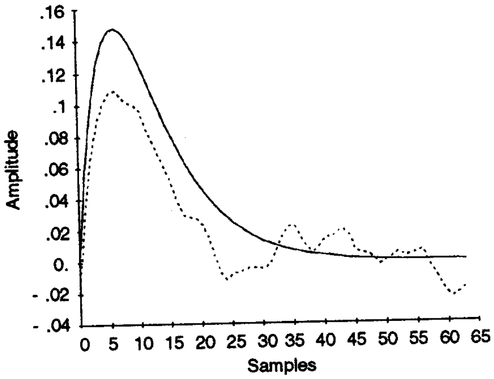


*Fig. 4. First-order kernel, true and cross-correlation estimate for Example 3. Solid line, true; dashed line, cross-correlation estimate.*

$$E = \frac{\overline{y^2_{i-1}[n]}}{\overline{y^2[n]}}$$

With $h_i[m]$ defined, calculate its output

$$u_i[n] = \sum_{m=0}^{R} h_i[m]\, x[n-m]$$

Next, a polynomial having input $u_i$ is best-fit to the residue $y_{i-1}$ over the interval $n = R,\ldots,N$. This determines the static nonlinearity in the cascade, and then the cascade output $z_i$ can be calculated. The new residue can be obtained from (30), and the process continued until the mse is acceptably small, or a given number of cascades have been added to the model, or tested. For example, before adding a given candidate for the (i+1)-th cascade, one may optionally require that

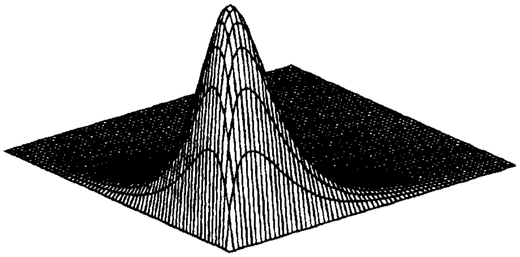


*Fig. 5. True second-order kernel for Example 3.*

$$\overline{z^2_{i+1}[n]} > \frac{4}{N-R+1}\, \overline{y_i^2[n]} \quad (33)$$

which can be obtained analogously to (25), and helps to prevent choosing unnecessary cascades which are merely fitting noise. Suppose that this optional test is used and the candidate cascade does not satisfy (33). Then randomly reselect $h_{i+1}[m]$ using a first-order crosscorrelation, or a *slice* of a second- or higher-order crosscorrelation to which discrete impulses are added or subtracted at diagonal values (as in (32)).

Notice that the cascade paths are obtained one at a time. Higher-order nonlinear systems are readily modelled by parallel cascade identification since they merely require estimation of higher-degree (and/or more) polynomials in the cascade paths. Indeed, the nonlinearities in the parallel cascade representation always appear as static functions. Hence their estimation is far faster than the higher-order crosscorrelation required by the Lee-Schetzen [16] method of estimating Wiener kernels. Note that only a *slice* of a crosscorrelation is required (in (31) and (32)) to define the first component of a cascade path, as opposed to complete crosscorrelation in the Lee-Schetzen approach.

Aside from the capability of rapidly modelling high-order nonlinear systems, parallel cascade identification

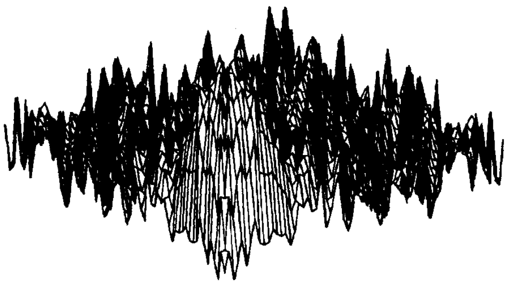


*Fig. 6. Cross-correlation second-order kernel estimate for Example 3.*

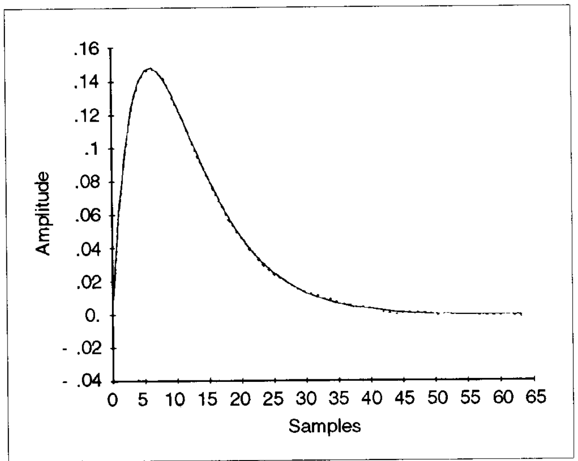


*Fig. 7. First-order kernel, true and parallel cascade estimate for Example 3. Solid line, true; dashed line, parallel cascade estimate.*

enables accurate kernel estimation even for systems with lengthy memory. This is done by rearranging the identified parallel cascade into an equivalent functional expansion whose kernels are then known. Since kernels may reveal much about information processing (see, for example, the retinal studies by Sakai, Naka, and colleagues [36]-[40]), accurate kernel estimation is important.

*Example 3*

To illustrate kernel estimation via the parallel cascade method, a second-order nonlinear system (L = 2 in (29)) with memory length 64 was stimulated by 5000 points of zero-mean, white, Gaussian noise. (For a second-order Volterra series, the corresponding Volterra and Wiener kernels of first- and second-order are equal.) One zero-order, 64 first-order, and 2080 distinct second-order kernel values were estimated by crosscorrelation [16] and parallel cascade [11]-[14] methods. As shown in Figs. 4, 5, and 6, there is considerable error between the crosscorrelation estimates and the true kernel values. The parallel cascade kernel estimates shown in Figs. 7 and 8 are significantly closer. The mean-square error of the parallel cascade representation is about 0.037%. The estimation can be made even more accurate by adding further cascade paths to the representation.

One reason for the noise in the crosscorrelation kernel estimates is that the record length (5000 points) is relatively short compared with the total number of distinct kernel values (2145), many of which have significant magnitude. Crosscorrelation estimates do approach the actual kernel values as the record length increases. However, in other comparative testing, crosscorrelation estimates did not attain the accuracy of parallel cascade estimates even when the record used for the crosscorrelation method was 100 times longer than that for parallel cascade. Note that, in the present example, attempting to estimate the kernels of the test system by direct least-squares estimation would entail the numerically formidable inversion of a symmetric 2145 by 2145 matrix (which is not Toeplitz nor near-Toeplitz so as to enable rapid inversion).

In fact, the parallel cascade method [11]-[14] has been successfully applied to kernel estimation of systems with memory length up to 150. Such kernels could not practically be measured by direct least-squares estimation since, for a second-order system, inversion of a 11,476 by 11,476 matrix would be required. Finally, the parallel cascade method does not require use of a Gaussian input, nor an input having special probability density or autocorrelation properties. Rather, the method can accurately model high-order nonlinear systems, as well as estimate kernels, for a wide variety of input excitation [13],[14].

## TIME-SERIES ANALYSIS

Let y[n], $n = 0,\ldots,N$, now represent given time-series data, and let (2) represent a parsimonious sinusoidal series representation which is to be developed for the time-series. The frequencies of the sinusoids in this representation will be chosen from a set of candidate frequencies $\omega_A$, $\omega_B$, .... These candidate frequencies need not be commensurate, nor integral multiples of a fundamental frequency. The selection of frequencies is carried out by fast orthogonal search, which was developed by Korenberg [4],[5] for modeling time-series data as well as for the difference equation modeling discussed above.

In the sinusoidal series represented by (2), note that (3) still holds, and for $i = 1,2,\ldots$

$$p_{2i-1}[n] = \cos\omega_i n$$

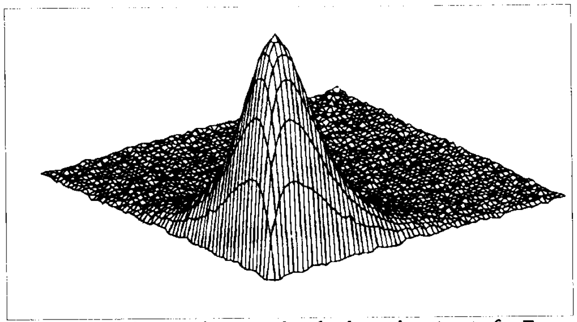


**Fig. 8. Parallel cascade second-order kernel estimate for Example 3.**

$$p_{2i}[n] = \sin\omega_i n$$

The $\omega_i$ can be selected by systematically searching through the set of candidate frequencies $\omega_A$, $\omega_B$, ... as discussed below. It can be shown [4],[5] that adding the i-th term pair

$$T_i = a_{2i-1}\,p_{2i-1}[n] + a_{2i}\,p_{2i}[n]$$

to the model of (2) decreases the mean-square error by

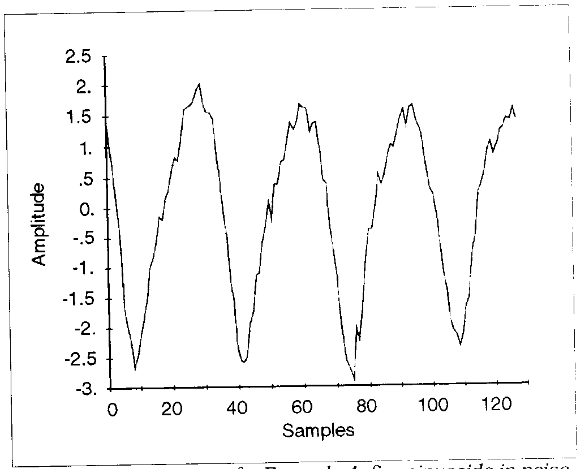


*Fig. 9. Noisy time-series for Example 4: five sinusoids in noise.*

the amount

$$Q_1[i] = G^2[2i-1]\, D\,[2i-1,2i-1] + G^2[2i]\, D\,[2i,2i] \qquad (34)$$

At the stage of adding the i-th term pair, $M = 2i$ in (2). To find $\omega_i$, $i \geq 1$, evaluate $Q_1[i]$ for each available (i.e., as yet unselected) candidate frequency. Choose the candidate frequency with largest $Q_1$ value, optionally subject to exceeding a minimum threshold level. For example, before choosing a candidate frequency for $\omega_i$, one may optionally require [27] that

$$G^2[M-1]\, D\,[M-1,M-1] + G^2[M]\, D\,[M,M] > \frac{4}{N+1}\left(\overline{y^2[n]} - \sum_{m=0}^{M-2} G^2[m]\, D\,[m,m]\right) \qquad (35)$$

which can be derived similarly to (25). If (35) is used, one should also require that the mse exceed a threshold level before continuing the model development. For shorter records, the factor $\frac{4}{N+1}$ (which corresponds to 95% confidence limits) on the right side of (35) can be replaced by the square of the critical correlation coefficient value for sample size $N+1$. Clearly, a factor corresponding to other than 95% confidence limits can be substituted on the right side of (35).

To evaluate $Q_1[i]$ in (34) for a given candidate for $\omega_i$, use the pseudocode from (10) to the line following (15), but only for $m \geq M-1$ (where $M = 2i$). This avoids repeating calculations previously performed at an earlier stage of searching. After selecting the candidate for $\omega_i$, again carry out the pseudocode (for $m \geq M-1$ ) for the chosen candidate. This is done to properly set the values of C[m], G[m], D[m,m] and A[m,r], $m = M-1$, $M$, $r = 0,\ldots, m-1$, prior to either searching for $\omega_{i+1}$, or terminating the process and using (16)-(18) to find the coefficients $a_m$ in (2). In using the pseudocode, note that the overbar on the right side of (11), (12), (13) and (14) now denotes the time-average from $n = 0$ to $n = N$. For example,

$$\overline{y[n]} = \frac{1}{N+1}\sum_{n=0}^{N} y\,[n]$$

Using the time-average from $n = 0$ to $n = N$, note that (24) will give the mean-square error for the sinusoidal series model in (2). Model development may be stopped when the model mean-square error is a sufficiently small percentage of the time-series variance, or if a preset limit on the number of model frequencies has been reached. Termination may also occur if no remaining candidate frequency can cause a reduction in mean-square error exceeding a specified threshold level. Finally, note that this method of modeling time-series data can also be applied with unequally-spaced or missing data.

Fast orthogonal search is capable of much finer frequency resolution than a conventional Fourier series analysis (see Example following). Moreover, in the well-known AR and maximum entropy methods, model order must be selected, and this is also true for the Prony and Pisarenko methods, which additionally require solution of a polynomial equation [41]-[44]. In fast orthogonal search, model order is automatically determined, and no polynomial equation must be solved.

A method published earlier, by Abraira and Ibarz [45], also scans a set of candidate frequencies to choose successively, via a best-fitness criterion, the significant frequencies for the sinusoidal series representation. Their method does not use an orthogonal approach, and the same frequency can be chosen more than once [45]. The final amplitude of the sinusoidal component at a selected frequency is the sum of the values determined each time the frequency is chosen (taking into account the phase), and is only an approximation to the correct amplitude. The searching can continue even if each of the correct frequencies has been selected once, so there is the possibility of incorrect frequencies being selected. Since each sinusoidal component selected is subtracted from the time series, this may introduce incorrect frequencies and undermine subsequent searching. How-

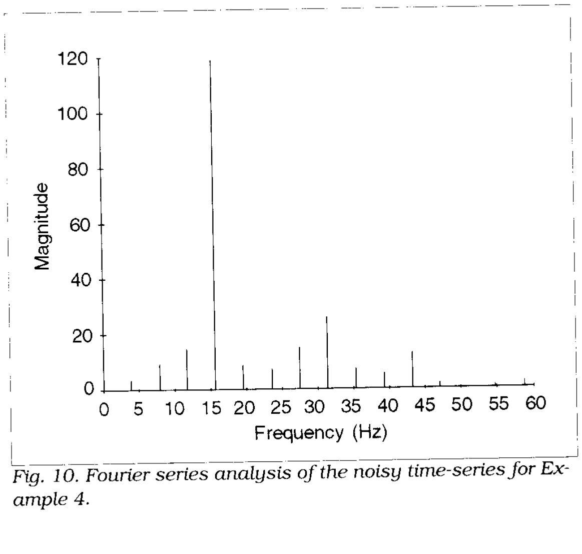


*Fig. 10. Fourier series analysis of the noisy time-series for Example 4.*

**TABLE I**

SINUSOIDAL ANALYSIS FOR EXAMPLE 4
(Five sinusoids in noise; SNR = 20 dB)

| True | | | Estimated | | |
|---|---|---|---|---|---|
| Frequency in Hz | Amplitude | Phase in Degrees | Frequency in Hz | Amplitude | Phase in Degrees |
| 5.0 | 0.1 | —68.0 | 5.0 | 0.1167 | —79.22 |
| 15.0 | 1.0 | 60.0 | 15.0 | 0.9307 | 58.69 |
| 15.5 | 1.0 | 75.0 | 15.5 | 1.0714 | 72.87 |
| 30.0 | 0.5 | 15.0 | 30.0 | 0.5020 | 15.30 |
| 44.0 | 0.2 | —45.0 | 44.0 | 0.1881 | —50.45 |

ever, the method offers greater resolution than the Fourier series approach, is not limited to use of commensurate frequencies, and does not require the time-series data to be equally-spaced.

Note that fast orthogonal search need select each correct frequency only once, and choosing an extra (incorrect) frequency during the searching does not introduce error provided the correct frequencies are eventually chosen. The algorithm will then yield negligible values for the sine and cosine amplitudes at the incorrect frequency.

*Example 4*

The improved resolution of fast orthogonal search over the conventional Fourier series approach is demonstrated on the 128-point test time-series shown in Fig. 9. The assumed sampling period is 2 msec, so that the sampling rate is 500 Hz. The test time-series consists of five sinusoids with additive noise; the sinusoids are indicated in Table I, and the additive noise is zero-mean, Gaussian noise such that the overall SNR is 20 dB. The SNRs for the individual sinusoids are as follows (frequency / SNR): 5 Hz / -5.4 dB, 15 Hz / 14.1 dB, 15.5 Hz / 14.1 dB, 30 Hz / 8.2 dB, 44 Hz / 0.3 dB. Without the additive noise the algorithm identifies all frequencies, amplitudes, and phase angles essentially without error, even though the number of frequencies present is not known a priori. With the additive noise present, the results are shown in Table I. Note that all five frequencies were exactly selected, however some error is noted in the amplitudes and the phase angles due to the noise.

Note that the frequency resolution of Fourier series analysis on the same data is 500 Hz / 128 = 3.91 Hz. The magnitude Fourier series, for comparison, is shown in Fig. 10; the frequency scale only goes to 60 Hz, rather than 250 Hz, to give an expanded scale; frequency components above 60 Hz are insignificant. The five most dominant frequencies present in the Fourier series are as follows, in descending order: 15.625 Hz, 31.25 Hz, 27.3438 Hz, 11.7188 Hz and 42.9688 Hz. However, as shown in Table I, of the five frequencies in the test time-series, two (namely, 15 Hz and 15.5 Hz) are separated by only 0.5 Hz. Note that fast orthogonal search was able to resolve these two frequencies, but that the Fourier series analysis was not. Appending zeros onto the data file or windowing the data before performing the Fourier series analysis does not improve the analysis. In this example, the frequency resolution of fast orthogonal search has been shown to be eight times better than Fourier series analysis (0.5 Hz versus 3.91 Hz).

In carrying out the fast orthogonal search, 100 candidate frequencies equally-spaced between 0.5 Hz and 50 Hz (inclusive) were searched. Higher resolution is possible (in other examples) by increasing the resolution of the candidate frequencies. For additional testing of the resolution of fast orthogonal search, see [46],[47]. Note that fast orthogonal search is not always as accurate as shown in this example, but in extensive testing has continually surpassed the Fourier series result. Moreover, in [5] fast orthogonal search was tested on a 64-point sample sequence used by Kay and Marple [42] to study several spectral estimation procedures. Fast orthogonal search exhibited significantly higher resolution than reported [42] for any of the studied procedures except for the spectral line variant of the Prony method. Compared with the latter method, fast orthogonal search showed equivalent accuracy in estimating frequencies and amplitudes, and much greater accuracy in estimation of phases.

## CONCLUSIONS

In this paper, some recent approaches to nonlinear system identification, ARMA modeling, and time-series analysis have been examined. Almost all of these methods are related by analysis methods based on an orthogonal search procedure. Sufficient detail and references have been furnished to enable ready implementation, and examples were provided to demonstrate superiority over established classical techniques.

In the case of ARMA modeling, the ARMA identification algorithm presented does not require a priori knowledge of, or assumptions about, the order of the system to be identified or signal to be modeled. A suboptimal, recursive, pairwise search of the orthogonal candidate data records is conducted, until a given least-squares criterion is satisfied. The basic ARMA systems identification algorithm has been extended for improved performance in the presence of additive noise, by the method of MA noise reduction. Example 1 illustrates the performance of the ARMA identification algorithm with noisy data.

The basic ARMA algorithm has also been extended for the signal modeling case (no access to input data). Example 2 illustrates the performance of the ARMA signal modeling algorithm.

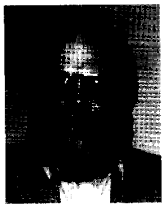
In the case of nonlinear systems modeling, discrete-time Volterra series has been stressed, or rather a more efficient parallel-cascade approach. The model is constructed by adding parallel paths (each consisting of the cascade of dynamic linear and static nonlinear systems). Kernel estimation via the parallel-cascade method is illustrated in Example 3.

In the case of time-series analysis, a non-Fourier sinusoidal series approach has been stressed. The relevant frequencies, corresponding amplitudes and phase angles, are estimated by an orthogonal search procedure. A search of the candidate sinusoids is conducted until a given mean-square criterion is satisfied. Time-series modeling via this method is illustrated in Example 4.

## Acknowledgment

This work was supported by the Natural Sciences and Engineering Research Council (Canada). For correspondence, contact either author.

**M.J. Korenberg** is a Professor in the Department of Electrical Engineering at Queen's University, Kingston, Ontario K7L 3N6, Canada. He received his B.Sc. (1965), M.Sc. (1968, Mathematics) and Ph.D. (1972, Electrical Engineering) degrees from McGill University, Montreal, Quebec, Canada. He has published extensively in the areas of nonlinear systems identification and time-series analysis, and is particularly interested in applications to biomedical modeling and signal processing. Other applications of interest relate to digital communications, such as identifying dispersive nonlinear data transmission channels.



**Larry D. Paarmann** received the B.S., M.S. and Ph.D. degrees, all in electrical engineering, from Northern Illinois University, the University of Illinois at Urbana, and Illinois Institute of Technology, respectively. He is currently an Associate Professor of Electrical Engineering at The Wichita State University. Previously, he was an Assistant Professor at Drexel University, and prior to that an instructor at Illinois Institute of Technology, and a member of the technical staff at IIT Research Institute. He has authored or coauthored over 25 papers in the areas of time-series modeling, systems identification, speech coding, and processing of speech for the hearing impaired. He is a member of the following societies of the IEEE: Signal Processing, Control Systems, Circuits and Systems, and Communications. He is also a member of the European Association for Signal Processing (EURASIP), the Biomedical Engineering Society, and the Acoustical Society of America.

## REFERENCES

[1] R. Haber and L. Keviczky, "Identification of Nonlinear Dynamic Systems," *IFAC Symp. Ident. Sys. Param. Est.*, pp. 79-126, 1976.

[2] S.A. Billings and I.J. Leontaritis, "Parameter Estimation Techniques for Nonlinear Systems," *6th IFAC Symp. Ident. Sys. Param. Est.*, Washington, DC, pp. 427-432, 1982.

[3] S.A. Billings and M.B. Fadzil, "The Practical Identification of Systems with Nonlinearities," *7th IFAC Symp. Ident. Sys. Param. Est.*, York, U.K., pp. 155-160, 1985.

[4] M.J. Korenberg, "Fast Orthogonal Identification of Nonlinear Difference Equation and Functional Expansion Models," Proc. 30th Midwest Symp. Cir. Sys., pp. 270-276, August 1987.

[5] M.J. Korenberg, "A Robust Orthogonal Algorithm for System Identification and Time-Series Analysis," *Biol. Cybern.*, Vol. 60, pp. 267-276, 1989.

[6] M.J. Korenberg and L.D. Paarmann, "An Orthogonal ARMA Identifier with Automatic Order Estimation for Biological Modeling," *Ann. Biomed. Eng.*, Vol. 17, pp. 571-592, 1989.

[7] M.J. Korenberg, "Orthogonal Identification of Nonlinear Difference Equation Models," *Proceed. 28th Midwest Symp. Cir. Sys.*, pp. 90-95, Louisville, KY, Aug. 1985.

[8] P.J.H. McIlroy, "Applications of Nonlinear Systems Identification," M.S. Thesis, Queen's University, Kingston, Ontario, Canada, 1986.

[9] M.J. Korenberg, S.B. Bruder, and P.J.H. McIlroy, "Exact Orthogonal Kernel Estimation from Finite Data Records: Extending Wiener's Identification of Nonlinear Systems," *Ann. Biomed. Eng.*, Vol. 16, pp. 201-214, 1988.

[10] M. Korenberg, "Identifying Nonlinear Difference Equation and Functional Expansion Representations: The Fast Orthogonal Algorithm," *Ann. Biomed. Eng.*, Vol. 16, pp. 123-142, 1988.

[11] M.J. Korenberg, "Statistical Identification of Parallel Cascades of Linear and Nonlinear Systems," *IFAC Symp. Ident. Sys. Param. Est.* (preprints), Washington, DC, pp. 580-585, 1982.

[12] M.J. Korenberg, "Functional Expansions, Parallel Cascades, and Nonlinear Difference Equations," in *Advanced Methods of Physiological System Modeling*, Vol. 1, V.Z. Marmarelis (ed.), U.S.C. Biomedical Simulations Resource, Los Angeles, pp. 221-240, 1987.

[13] M.J. Korenberg, "A Rapid and Accurate Method for Estimating the Kernels of a Nonlinear System with Lengthy Memory," *15th Biennial Symp. Comm.*, Dept. E.E., Queen's Univ., Kingston, Ontario, Canada, pp. 57-60, June 3-6, 1990.

[14] M.J. Korenberg, "Parallel Cascade Identification and Kernel Estimation for Nonlinear Systems," *Ann. Biomed. Eng.*, Vol. 19, 1991.

[15] N. Wiener, *Nonlinear Problems in Random Theory*, New York, Wiley, 1958.

[16] Y.W. Lee and M. Schetzen, "Measurement of the Wiener Kernels of a Nonlinear System by Crosscorrelation," *Int. J. Contr.*, Vol. 2, pp. 237-254, 1965.

[17] A.S. French and E.G. Butz, "Measuring the Wiener Kernels of a Nonlinear System Using the Fast Fourier Transform Algorithm," *Int. J. Contr.*, Vol. 17, pp. 529-539, 1973.

[18] G. Palm and T. Poggio, "Stochastic Identification Methods for Nonlinear Systems: An Extension of the Wiener Theory," *SIAM J. Appl. Math.*, Vol. 34, pp. 524-534, 1978.

[19] P.Z. Marmarelis and V.Z. Marmarelis, *Analysis of Physiological Systems. The White Noise Approach*, Plenum Press, New York, 1978.

[20] P.Z. Marmarelis and K.-I. Naka, "White-Noise Analysis of a Neuron Chain: An Application of the Wiener Theory," *Science*, Vol. 175, pp. 1276-1278, 1972.

[21] M. Sakuranaga, S. Sato, E. Hida and K.-I. Naka, "Nonlinear Analysis: Mathematical Theory and Biological Ap-

plications," *CRC Crit. Rev. Biomed. Eng.*, Vol. 14, pp. 127-184, 1986.

[22] A.A. Desrochers, "On an Improved Model Reduction Technique for Nonlinear Systems," *Automatica*, Vol. 17, pp. 407-409, 1981.

[23] G.H. Golub and C.F. Van Loan, *Matrix Computations*, 2nd ed., Johns Hopkins Univ. Press, Baltimore, 1989.

[24] M.J. Korenberg, "A Fast Orthogonal Search Method for Biological Time-Series Analysis and System Identification," *IEEE Inter. Conf. Sys. Man, Cybern.*, Cambridge, MA, pp. 459-465, 1989.

[25] A.A. Desrochers and S. Mohseni, "On Determining the Structure of a Non-Linear System," *Int. J. Contr.*, Vol. 40, pp. 923-938, 1984.

[26] M.J. Korenberg and P.H. Wittke, "Adaptive Identification of Dispersive Nonlinear Data Transmission Channels," *IEEE Inter. Conf. Communications*, Philadelphia, PA, pp. 939-945, 1988.

[27] M.J. Korenberg, "Fast Orthogonal Algorithms for Nonlinear System Identification and Time-Series Analysis," in *Advanced Methods of Physiological System Modeling*, Vol. 2, V.Z. Marmarelis (ed.), Plenum Press, New York, pp. 165-177, 1989.

[28] J.R. Rice, "A Theory of Condition," *SIAM J. Numer. Analysis*, Vol. 3, pp. 287-310, 1966.

[29] I.S. Konvalinka and M.R. Matausek, "Simultaneous Estimation of Poles and Zeros in Speech Analysis and ITIF-Iterative Inverse Filtering Algorithm," *IEEE Trans. Acoust., Speech, Signal Process.*, Vol. ASSP-27, pp. 485-492, Oct. 1979.

[30] L.D. Paarmann and M.J. Korenberg, "Estimation of the Parameters of an ARMA Signal Model Based on an Orthogonal Search," in press, *IEEE Trans. Auto. Contr.*, 1991.

[31] M. Frechet, "Sur Les Fonctionnelles Continues," *Ann. Sci. Ecole Normal Sup.*, Vol. 27, pp. 193-219, 1910.

[32] J. Dieudonne, *Foundations of Modern Analysis*, Springer, Berlin, Heidelberg, New York, 1976.

**[33] G. Palm, "On the Representation and Approximation of Nonlinear Systems, Part II: Discrete Time," *Biol. Cybern.*, Vol. 34, pp. 49-52, 1979.**

[34] M.J. Korenberg, "Some New Approaches to Nonlinear System Identification and Time-Series Analysis," *12th Annual IEEE Inter. Conf. Eng. Med. Biol. Society*, Philadelphia, PA, pp. 20-21, 1990.

[35] A.G. Bose, "A Theory of Nonlinear Systems," Tech. Rept. No. 309, Res. Lab. Elect., M.I.T., Cambridge, MA, 1956.

[36] H.M. Sakai and K.-I. Naka, "Dissection of the Neuron Network in the Catfish Inner Retina. I. Transmission to Ganglion Cells," J. Neurophysiol., Vol. 60, pp. 1549-1567, 1988.

[37] H.M. Sakai and K.-I. Naka, "Dissection of the Neuron Network in the Catfish Inner Retina. II. Interactions Between Ganglion Cells," *J. Neurophysiol.*, Vol. 60, pp. 1568-1583, 1988.

[38] M.J. Korenberg, H.M. Sakai and K.-I. Naka, "Dissection of the Neuron Network in the Catfish Inner Retina. III. Interpretation of Spike Kernels," *J. Neurophysiol.*, Vol. 61, pp. 1110-1120, 1989.

[39] H.M. Sakai and K.-I. Naka, "Dissection of the Neuron Network in the Catfish Inner Retina. IV. Bidirectional Interactions Between Amacrine and Ganglion Cells," *J. Neurophysiol.*, Vol. 63, pp. 105-119, 1990.

[40] H.M. Sakai and K.-I. Naka, "Dissection of the Neuron Network in the Catfish Inner Retina. V. Interactions Between NA and NB Amacrine Cells," *J. Neurophysiol.*, Vol. 63, pp. 120-130, 1990.

[41] G.E.P. Box and G.M. Jenkins, *Time Series Analysis: Forecasting and Control*, Oakland, CA, Holden-Day, 1976.

[42] S.M. Kay and S.L. Marple, "Spectrum Analysis - A Modern Perspective," *Proc. IEEE*, Vol. 69, pp. 1380-1419, 1981.

[43] N.C. Mohanty, Random Signals Estimation and Identification. Analysis and Applications, New York, Van Nostrand, 1986.

[44] S.M. Kay, *Modern Spectral Estimation: Theory and Application*, Englewood Cliffs, NJ, Prentice-Hall, 1988.

[45] V. Abraira and J.M. Ibarz, "Spectral Estimation of Temporal Series at Unequal Intervals," *Computers Biomed. Res.*, Vol. 19, pp. 203-212, 1986.

[46] M.J. Korenberg and L.D. Paarmann, "Applications of Fast Orthogonal Search: Time-Series Analysis and Resolution of Signals in Noise," *Ann. Biomed. Eng.*, Vol. 17, pp. 219-231, 1989.

[47] M.J. Korenberg, "Biological Time-Series Analysis Via a Fast Orthogonal Search," *Proc. 15th Ann. Northeast Bioeng. Conf.*, pp. 149-150, 1989.

## Editor's Message *(continued)*

the column was used as a basis for a class discussion. In many others, no doubt, it was used to light the spring barbecue. Whatever the opinion, it is clear that we are a thoughtful and caring society. We are not a rigid monolith, but a broad spectrum of people who happen to share a common professional expertise. We share a commitment and interest in signal processing, but when the subject ranges beyond FFTs and HMMs, our thoughts, experiences, ideas, talents, concerns, beliefs, and cultures are as diverse as humanity itself — which, after all, is what we represent as a professional society. In short, your letters confirm that there is no typical "Engineer," and you may show this to the next person you hear poke fun at our profession (or better yet, tape it to your pocket protector).

Among the letters received was one comment that has caused me to consider once again an important missing feature of this publication. "I am writing in response to your editorial," began one correspondent, "... even though I noticed (to my surprise) that your magazine has no section for letters to the editor. You may want to add such a feature in the interest of fairness." He's right. There should be a column for letters. Not only to express points of view on editorial content, but for any reason that a society member would wish to share ideas of general interest to the readership. Dissemination of views and information about signal processing is, after all, the primary mission of this publication. Therefore, at the ICASSP '91 meeting of the Society's Publication Board in May, I requested that the institution of a letters column be considered. I am pleased to report that approval has been granted. Between now and the October issue, we will formulate some guidelines for the column.

In the meantime, "keep those cards and letters coming". Your comments and criticisms of the magazine are always appreciated. But please don't write to ask whether you can have my quick-draw calculator holster. It goes too nicely with my LED readout tie bar and my hexadecimal VLSI wristwatch.

**Jack Deller**