

CPB 00642

A computer program for linear nonparametric and parametric identification of biological data

Susan A.S. Werness and David J. Anderson *

Bioengineering Program, Kresge Hearing Research Institute, University of Michigan Medical School, 1301 E. Ann St., Ann Arbor, MI 48109, USA

A computer program package for parametric and nonparametric linear system identification of both static and dynamic biological data, written for an LSI-11 minicomputer with 28 K of memory, is described. The program has 11 possible commands including an instructional help command. A user can perform nonparametric spectral analysis and estimation of autocorrelation and partial autocorrelation functions of univariate data and estimate nonparametrically the transfer function and possibly an associated noise series of bivariate data. In addition, the commands provide the user the means to derive a parametric autoregressive moving average model for univariate data, to derive a parametric transfer function and noise model for bivariate data, and to perform several model evaluation tests such as pole-zero cancellation, examination of residual whiteness and uncorrelatedness with the input. The program, consisting of a main program and driver subroutine as well as six overlay segments, may be run interactively or automatically.

Linear system identification Minicomputer Parameter estimation Transfer functions Noise models
Integrated autoregressive moving average models Spectral analysis Autocorrelations Partial autocorrelations

1. INTRODUCTION

Linear system identification techniques used by control engineers have been of widespread interest to biologists and medical scientists [1-3]. These techniques have been applied in various ways to increase understanding of these systems and in the development of diagnostic tests. For example, nonparametric transfer functions derived by Fourier methods have been invoked in developing an understanding of stretch reflex dynamics [4], of eye tracking dynamics [5], and of postural control mechanisms [6]. Nonparametric spectral analysis has been useful for viewing biological noise

processes in the frequency domain [7-9].

Concise parametric models have been employed to evaluate parameters not directly accessible such as those relating to the uptake, distribution, and elimination of drugs [3,10], or physiological metabolites [11-14], and to investigate physiological or biochemical structures [3,15]. Once their structures and/or coefficients have been determined, these parametric models can be used to simulate responses to various kinds of inputs, such as drugs [16], or nervous stimulation [17]. They can be evaluated from mathematical points of view such as control system theory in order to gain insights concerning the underlying biological systems. These insights can aid in the design of physiological experiments. They can also be useful

* To whom correspondence should be addressed

to a control engineer who develops devices to control or to be controlled by these physiological processes [18]. Mathematical analysis of parametric models has clinical diagnostic applications [19]. For example, discriminant analysis techniques have been applied to features of parametric models of EEG records such as poles and zeros in the development of mathematical diagnostic procedures [20]. These poles and zeros economically characterize the dominant spectral content of the EEG records.

Noise exists in biological systems from the membrane level to the social level. Although biological noise is interesting to model in its own right [21–23], it can complicate the assessment of a system's dynamics. While disturbing and measuring the response of a system with correlated internal noise is a necessary step for assessing the dynamics of such a system, the coefficient estimates of a resulting parametric transfer function will be biased unless the noise is taken into account during the parameter estimation [24,25]. In order to do so, the characteristics of the noise must be already known or a parametric noise model must be evaluated simultaneously with the transfer function. Also, since noise is an inherent part of some physiological systems, a comprehension of this noise is a prerequisite to the detailed understanding of the system's dynamics [5,26]. Thus, it is important to be able to deal with a correlated noise component when deriving linear models of physiological systems.

The program package to be described here, IDENT, has as one of its most important characteristics the ability to estimate the parameters of a noise model simultaneously with those of a transfer function. This program package was developed in FORTRAN IV for an LSI-11 computer with 28 K of memory for the purpose of obtaining concise parametric models of subjects' waist and shoulder position responses to translational postural disturbances applied to their base of support. Since the postural responses to the disturbances were accompanied by low frequency postural sway noise, it was necessary to incorporate a noise component into the parametric model. The algorithm employed is a modification of the extended Kalman filter described in Panuska [27]. Since this

algorithm is recursive in nature, it can be modified to track time varying parameters of relatively long data series. For the posture data, it was used to find the time invariant parameters of relatively short input–output series (< 350 data points). The orders of both the transfer function and noise model can be arbitrary. The parameters of a transfer function with no noise component or of an ARIMA (integrated autoregressive moving average) model of a univariate time series may be estimated. An ARIMA model contains poles, zeros, and/or differencing terms [28]. Asymptotically, the parameter estimates of the extended Kalman filter used here converge to a stationary point of the same loss function used in the method of maximum likelihood [27,29] and are thus asymptotically unbiased and efficient [25].

Overlaid with the parameter estimator are subroutine packages for the purpose of performing nonparametric (Fourier) analysis of either bivariate input–output data or of univariate time series. Nonparametric transfer functions and impulse responses can be derived for input–output data. These nonparametric results for the posture data were used to obtain initial estimates of model order. The mean, variance, normalized autocovariance, and partial autocorrelation functions as well as a smoothed estimate of the spectrum can be evaluated for a univariate time series.

Other overlay segments contain subroutines used for calculating diagnostic criteria for the evaluation of the parametric models. These criteria include the normalized autocovariance function, partial autocorrelation function, and variance of the residual or error series. In addition, a normalized cross-covariance function of the residuals and the prewhitened input is calculated for models which include a transfer function component. The poles and zeros of the parametric models are extracted and can be checked for near cancellation or insignificance. The parametric transfer function is evaluated at $e^{-j\omega}$ and can be compared to a nonparametric one derived with Fourier methods. These criteria can be used in a model selection process or simply to evaluate a model in cases where the model structure is known a priori.

The modeling package described above may be run interactively or in an automatic mode in which

commands to the program are obtained from a previously constructed command file. These command files may be generated with any system editor or with an accompanying FORTRAN IV program, COMGEN. Although IDENT was developed for the LSI-11 computer with an RT-11 version 4 operating system, the only characteristics which are system and data specific are those for opening, closing, and reading from data files. Thus, the package should be transferable with a user-supplied main program.

2. METHODS AND THEORY

2.1. Parameter estimator

The parameter estimation scheme is a linearized or extended Kalman filter [27]. A Kalman filter is a recursive one in which estimates of the 'state' are updated by amounts which depend upon a Kalman gain matrix and the size of the error with which the actual measurement data are approximated by a function of the state estimates. The Kalman gain is a function of the measurement noise which is assumed to exist in the actual output data and is calculated so that the state estimates, which are random processes, result in a minimum error variance. In this parameter estimation problem, the state vector is the set of parameters. The linearization occurs in the observation matrix, or the function of the state estimates by which the measured data are approximated since in general, the measured data are nonlinear functions of the parameter estimates.

The parameter estimation scheme is used to evaluate the parameters of the following model:

$$A(z)y(t) = z^{-d}B(z)x(t) + [C(z)/F(z)]e(t) \quad (1)$$

where z^{-1} is the backward shift operator and

$$A(z) = 1 + a_1z^{-1} + \dots + a_nz^{-n}$$

$$B(z) = b_0 + b_1z^{-1} + \dots + b_mz^{-m}$$

$$C(z) = 1 + c_1z^{-1} + \dots + c_qz^{-q}$$

$$F(z) = 1 + f_1z^{-1} + \dots + f_pz^{-p}$$

$y(t)$ is the output sequence

$x(t)$ is the input sequence

$n, m, q,$ and p are the model orders

d is a deadtime factor or delay term

The sequence $e(t)$ is a white noise sequence having zero mean and a covariance:

$$E[e(k)e(l)^T] = \Lambda\delta_{kl} \quad (2)$$

To facilitate the description of the estimation algorithm, the model of equation (1) is written as:

$$y(t) = \psi^T(t)\theta(t) + e(t) \quad (3)$$

with

$$\begin{aligned} \psi^T(t) &= [-y(t-1)\dots -y(t-n), x(t)\dots x(t-m), \\ &e(t-1)\dots e(t-q), -\epsilon(t-1)\dots -\epsilon(t-p)] \\ \theta^T(t) &= [a_1, \dots, a_n, b_0, \dots, b_m, c_1, \dots, c_q, f_1, \dots, f_p] \end{aligned}$$

$\epsilon(t)$ is a sequence defined by:

$$\epsilon(t) = C(z)/F(z)e(t)$$

Equation (3) can be considered to be an observation equation of a state representation of system (1). The corresponding state equation is:

$$\theta(t+1) = \theta(t) \quad (4)$$

Now, in reality, the sequences $e(t)$ and $\epsilon(t)$ are not available and must be estimated:

$$\hat{\epsilon}(t) = A(z)y(t) - B(z)x(t) \quad (5)$$

$$\hat{e}(t) = y(t) - \psi^T(t)\theta(t)$$

The extended Kalman filter algorithm to solve for the states, $\theta(t)$, of the above system now follows. To make use of the minimum variance formulation of the Kalman filter, an estimator of

the covariance Λ is coupled with the filter equations.

$$\hat{\theta}(t+1) = \hat{\theta}(t) + K(t)[y(t) - \psi^T(t)\hat{\theta}(t)]$$

$$\hat{\theta}(0) = \hat{\theta}_0 \quad (6)$$

$$K(t) = P(t)G^T(t)[G(t)P(t)G^T(t) + \Lambda(t+1)]^{-1} \quad (7)$$

$$P(t+1) = P(t) - K(t)[G(t)P(t)G^T(t) + \Lambda(t+1)]K^T(t)$$

$$P(0) = P_0 \quad (8)$$

$$\hat{\Lambda}(t+1) = \hat{\Lambda}(t) + 1/(t+1)[\hat{e}(t)\hat{e}^T(t) - \hat{\Lambda}(t)]$$

$$\hat{\Lambda}(0) = \Lambda_0 \quad (9)$$

where

$$G(t) = \delta/\delta\theta\psi(t)_{\theta_1=\hat{\theta}(t)} = -\delta/\delta\theta\hat{e}(t)_{\theta_1=\hat{\theta}(t)} \quad (10)$$

With the exception of the estimator for Λ , this extended Kalman filter algorithm is a special case of that found in many places [25,27,30,31]. The inclusion of the estimator for Λ (eq. (9)) is due to Panuska [27].

Equation (6) is commonly called the state estimate update equation [30]. $K(t)$, the Kalman gain, is updated in eq. (7). $P(t)$ is the error covariance matrix of the state estimates and eq. (8) provides the error covariance update.

The following modifications are applied to the algorithm outlined in eq. (6–10):

- (1) Deadtimes are included in the parameter estimation problem simply by shifting the output vector, $y(t)$, a specified number of places to the right.
- (2) Since the initial values for $\hat{\theta}(t)$, $P(t)$, and $\hat{\Lambda}$ (assigned to be equal to 0) are only convenient estimates, it is helpful to weigh later estimates more than early ones [25,32,33]. If, in the following weighting window, λ_0 and $\lambda(0)$ are

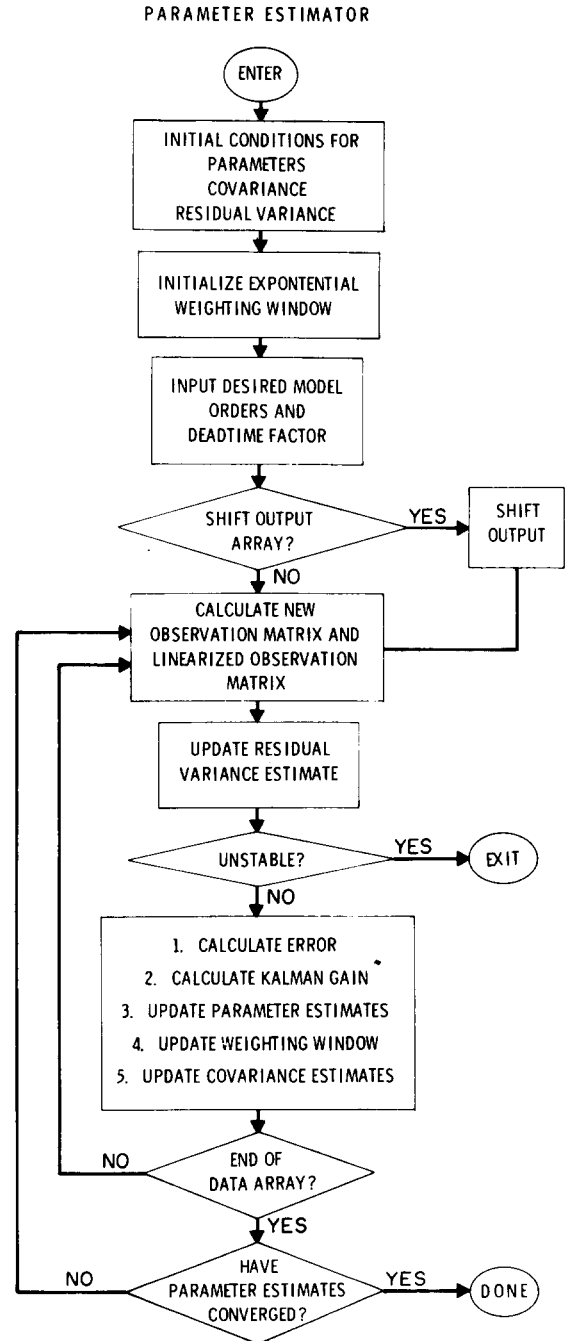


Fig. 1. Flowchart of the parameter estimation algorithm.

chosen to be close to 1, $\lambda(t)$ will tend to 1 exponentially.

$$\lambda(t+1) = \lambda_0\lambda(t) + (1 - \lambda_0) \quad (11)$$

- (3) For estimation of time invariant parameters of short time series, improvement of the estimates results from running the data through the filter several times, each time using as initial values the values obtained at the end of the previous run. The iteration stops when all the parameter estimates change by less than a specified criterion, .01. The parameter estimation algorithm with its modifications is flow charted in Fig. 1.

Until recently, the asymptotic properties of recursive estimators have only been studied by means of simulations since recursive estimators are nonlinear time varying stochastic difference equations in which the error estimates depend on all previous estimates and hence are difficult to analyze analytically [25,34]. However, by use of an associated ordinary time invariant differential equation, it is possible to show that the parameter estimates of the extended Kalman filter used here converge asymptotically to a stationary point of the same loss function used in the method of maximum likelihood [27,29]. Thus, asymptotically, these extended Kalman filter parameter estimates possess the same properties as maximum likelihood parameter estimates, they are asymptotically unbiased and efficient [20].

2.2. Nonparametric calculations

2.2.1. Univariate time series

A consistent estimator of the mean of a correlated series, $e(k)$, $k = 1, \dots, N$ is given as follows [35]:

$$\hat{m} = 1/N \sum_{k=1}^N e(k) \quad (12)$$

An asymptotically unbiased and efficient estimator for the variance of such a series is [28,36]:

$$\hat{V} = 1/N \sum_{k=1}^N (e(k) - \hat{m})^2 \quad (13)$$

An asymptotically unbiased and efficient esti-

imator of the normalized autocovariance function is [35] *:

$$\hat{r}(t) = 1/N \left(\sum_{k=1}^{N-t} [e(k) - \hat{m}][e(k+t) - \hat{m}] \right) / \hat{V} \quad (14)$$

Partial autocorrelations of a process may be estimated by fitting successively autoregressive processes of orders 1, 2, 3, ... by least squares, and picking out the estimates of the last coefficient of each stage. See Box and Jenkins [28] for a detailed explanation of the partial autocorrelation function. IDENT makes use of an approximate recursive method for calculation of these partial normalized autocovariance function estimates from the autocorrelation estimates [37]:

$$\hat{p}a_{t+1,j} = \hat{p}a_{tj} - \hat{p}a_{t+1,t+1} \hat{p}a_{t,t-j+1} \quad (15)$$

$$\hat{p}a_{t+1,t+1} = \hat{r}_{t+1} - \sum_{j=1}^t \hat{p}a_{tj} \hat{r}_{t+1-j} / 1 - \sum_{j=1}^t \hat{p}a_{tj} r_j$$

$$j = 1, 2, \dots, t$$

The estimates, $\hat{p}a_{t+1,t+1}$ are the partial autocorrelation estimates.

The normalized autocovariance and partial autocorrelation function estimates can be used to identify the structure of a univariate time series [28]. For example, it can be shown that the normalized autocovariance function of a moving average process of order q ($MA(q)$) is zero after the q -th lag. Also, the partial autocorrelation function of an autoregressive process of order p ($AR(p)$) is zero after the p -th lag. It has been demonstrated by Bartlett [38] that the standard deviations of the normalized autocovariance function of a $MA(q)$ after lag q can be approximated by:

$$\begin{aligned} \hat{\sigma}(r(k)) & \approx 1/N^{1/2} (1 + 2[r^2(1) + r^2(2) + \dots + r^2(q)])^{1/2} \\ & \text{for } k > q \end{aligned} \quad (16)$$

* In [28], [35] and others, eq. (14) is referred to as an estimate of the autocorrelation function. The term, normalized autocovariance, is consistent with engineering usage.

Quenouille [39] showed that the standard deviation of the partial autocorrelation function of an $AR(p)$ process after lag p is approximately:

$$\delta[\hat{\rho}_a(k)] \approx 1/N^{1/2} \quad k > p \quad (17)$$

According to Fuller [35], the distribution of an normalized autocovariance or partial autocorrelation coefficient whose theoretical value is zero is approximately Normal. Thus, one can informally determine whether these estimates beyond a certain lag are essentially zero. It can also be shown that the estimated normalized autocovariance function of a nonstationary process fails to damp out quickly, indicating that perhaps differencing of the series is warranted [28].

It is useful to point out some facts concerning the distributions of the normalized autocovariance function estimates of a white noise series. Fuller [35] demonstrates that the covariance matrix of the autocorrelation function of a white process is a diagonal one and thus that the following function is distributed as a chi square variable with L degrees of freedom:

$$Q_L = N \sum_{t=1}^L \hat{r}^2(t) \quad (18)$$

This result has been exploited to devise a 'portmanteau' test for residual whiteness to be used in evaluating models [40]. In IDENT is calculated another statistic to be used to test for serial correlation of a time series, the Durbin-Watson statistic [41,42]:

$$DW = 1/N \sum_{j=2}^N (e(t) - e(t-1))^2 / \hat{V} \quad (19)$$

The expected value of this statistic is equal to 2.0 for an uncorrelated series.

Oscillatory or seasonal series are more usefully viewed in the frequency domain. Figure 2 illustrates the methodology used in obtaining a smoothed estimate of the spectrum of a series. First, since the application of a rectangular window to a time series results in much side lobe leakage in the frequency domain, a tapered time domain

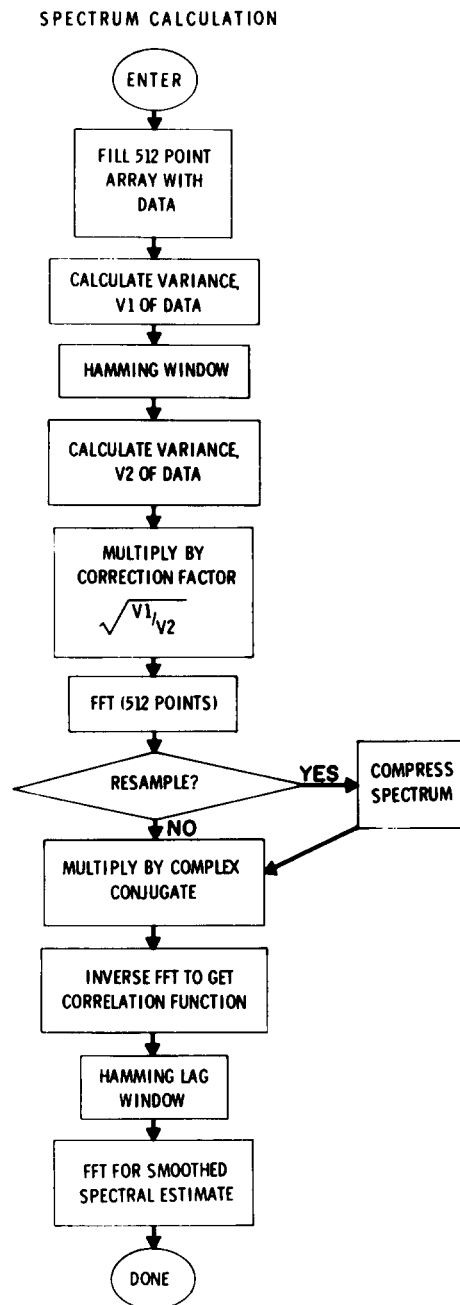


Fig. 2. Flowchart of the nonparametric spectral estimation algorithm.

window, the Hamming window, is applied to the data before entering the frequency domain via Fourier transforming [36,43]. For an N point se-

quence, this Hamming window is expressed:

$$W_H(m) = 0.54 + 0.46\cos(\pi m/N), m = 1, 2, \dots, N \quad (20)$$

Also, in order to avoid having the Fourier transform overwhelmed by a large amount of very low frequency energy, regression lines are fitted to the time domain sequences and then removed before the Fourier transform is obtained. The Fourier transform is calculated by the efficient FFT method [44]. Since spectral estimates are generally inefficient, possessing a high variance even for large data sequences, smoothing procedures are applied [36,45]. First, the windowed sequences are optionally resampled in the frequency domain to result in a Nyquist frequency (1/2 of the sampling rate) that is 1/2, 1/4, or 1/8 of the original Nyquist frequency, in order to eliminate unnecessary computational time. An intermediate estimate of the spectrum is formed by multiplying by the complex conjugate of the Fourier transform. An estimate of the autocorrelation function is obtained by inverse transforming. A Hamming lag window of either 1/2 or 1/4 the length of the autocorrelation function is applied and then a smoothed spectral estimate results with transformation back to the frequency domain.

2.2.2. Transfer function and impulse response estimation with correlated noise series identification

Figure 3 summarizes the methods by which the nonparametric transfer function and impulse response are estimated and an approximation of any accompanying noise series is obtained. The nonparametric transfer function results simply from division of the cross spectrum of the input and output time series by the input spectrum [33,35]:

$$H(e^{j\omega}) = S_y(e^{j\omega})S_x(e^{-j\omega})/|S_x(e^{j\omega})|^2 \quad (21)$$

$$\omega = 1, 2, \dots, 512$$

where:

$H(e^{j\omega})$ = the transfer function

$S_y(e^{j\omega})$ = the Fourier transform of the output series

ESTIMATION OF NONPARAMETRIC TRANSFER FUNCTION, IMPULSE RESPONSE, AND NOISE SERIES

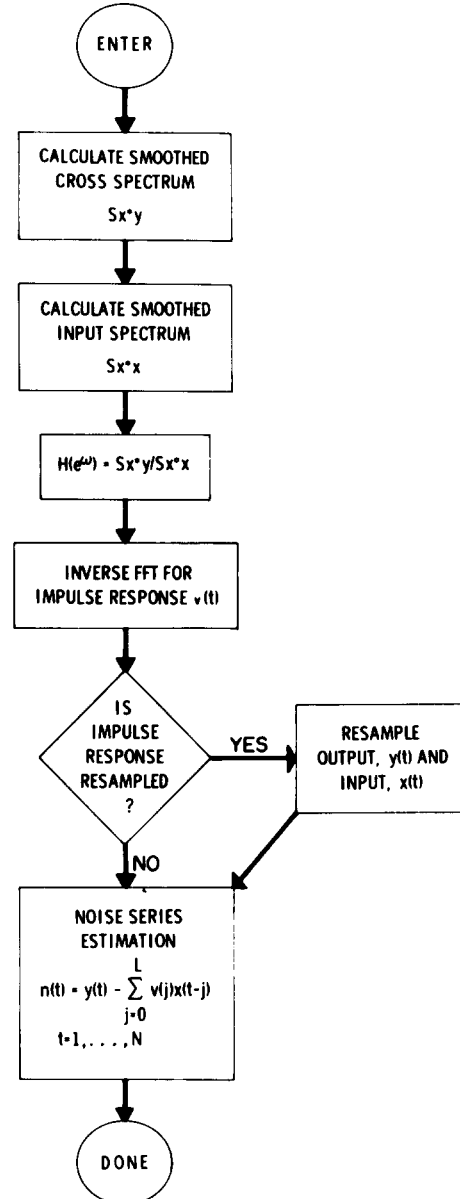


Fig. 3. Flowchart of the nonparametric transfer function and impulse response estimation algorithms.

$S_x(e^{-j\omega})$ = the complex conjugate of the input series' transform.

An impulse response of length L can be accurately obtained from the inverse transform of this

transfer function estimate as long as the data sequences used in the transfer function calculation are at least L points shorter than the size of the Fourier transform used [36]. If desired, an estimate of the noise accompanying the response to the input can be obtained using the estimated coefficients of the impulse response:

$$\hat{\nu}(t)y(t) - \sum_{j=0}^L \hat{\nu}(j)x(t-j), t = 1, \dots, N \quad (22)$$

where:

$\hat{\nu}(t)$ = the noise series estimate

$y(t)$ = the output sequence

$\hat{\nu}(j)$, $j = 0, \dots, L$, is the impulse response estimate, cut off after lag L (when the impulse response has damped out)

$x(t)$ is the input sequence.

This series, $\nu(t)$, corresponds to $C(Z)/[A(Z)F(Z)]e(t)$ of eq. (1) and is subjected to the univariate nonparametric analysis described above. The statistical properties of spectral and transfer function estimates are treated in Jenkins and Watts [45]. These estimates are inefficient in general, and used primarily to obtain model order information.

One can obtain rough guidelines of the transfer function model orders from either the nonparametric impulse response or transfer function. Box and Jenkins [28] describe in detail model order selection with the use of the impulse response. However, occasionally the impulse response estimate is less useful than the transfer function for this purpose. For example, it is possible that the impulse response coefficient estimates are small in comparison to their standard deviation. Also, if the frequency range of the input is narrow compared to the Nyquist frequency, the indeterminacy of the transfer function estimate in the higher frequency ranges causes the impulse response estimate to be quite poor. Of course, in these cases any estimate of the noise series will also be poor. From the transfer function estimate, one can obtain an inkling of at least the number of poles in the transfer function. For example, one peak in the transfer function at a frequency other

than 0 suggests the presence of a complex pole pair. Two such peaks suggest the presence of four poles. If a peak occurs at 0 Hz, then 1 or 2 real poles may be present.

2.3. Model evaluation criteria

The most difficult task in the development of a parametric model is the selection of the final model. Parametric model identification usually follows the iterative procedure outlined in Fig. 4, unless the model structure is already known. In other cases, usually several model selection tests are employed simultaneously in order to minimize the possibility of an incorrect model choice. The tests used and the relative weights applied to these tests depends upon the use of the model. Discussions of the subject appear in several places [24,26,28,36,46–51]. Outlined here are the calculations performed in IDENT and their use in model selection. Other tests could be added easily.

The residuals or unpredictable parts of the output data are used in several model evaluation tests. These residuals are calculated as follows [6]:

$$\hat{\epsilon}(t) = y(t) - \psi^T(t)\hat{\theta}(t) \quad t > n + d + p + 1 \quad (23)$$

where:

$y(t)$, $t = 1, 2, \dots, N$ is the output time series

$$\begin{aligned} \psi^T(t) = & [-y(t-1), \dots, \\ & -y(t-n), x(t), \dots, x(t-m), \\ & \times \hat{\epsilon}(t-1), \dots, \hat{\epsilon}(t-q), -\hat{\epsilon}(t-1), \dots, \\ & -\hat{\epsilon}(t-p)] \end{aligned}$$

$x(t)$, $t = 1, 2, \dots, N$ is the input time series

$$\hat{\epsilon}(t) = \hat{A}(z^{-1})y(t) - \hat{B}(z^{-1})x(t)$$

$$t > n + d + 1$$

$$\hat{\theta}^T(t) = [\hat{a}_1, \dots, \hat{a}_n, \hat{b}_0, \dots, \hat{b}_m, \hat{c}_1, \dots, \hat{c}_q, \hat{f}_1, \dots, \hat{f}_q]$$

If the parametric model is adequate, this estimated residual series should satisfy the assumption

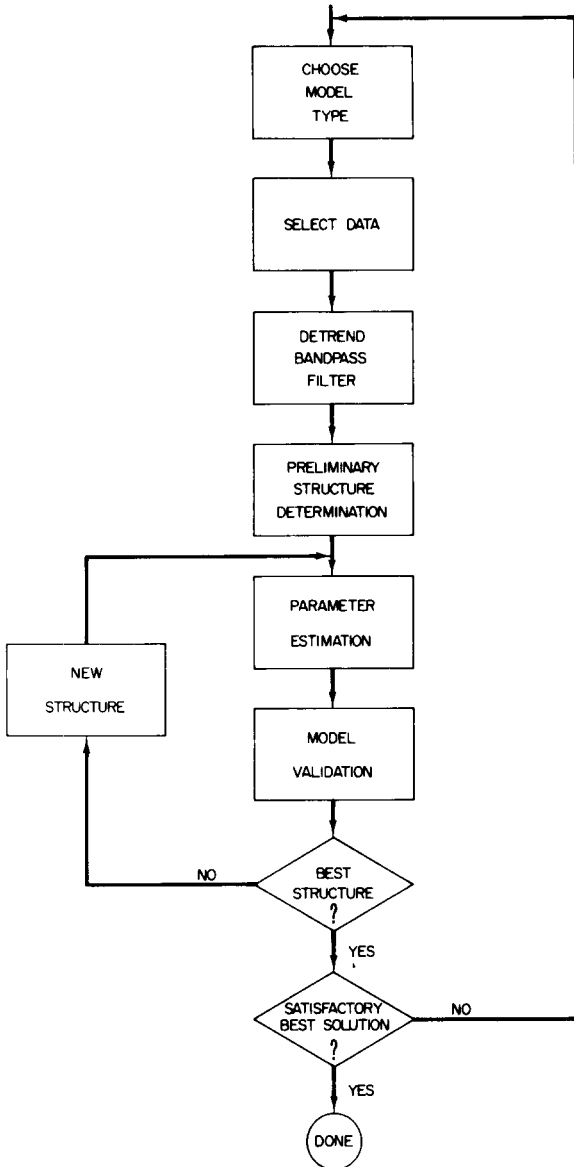


Fig. 4. Block diagram of parametric identification methodology.

of being uncorrelated. The univariate nonparametric analysis applied to the residuals provides information for checking these assumptions. The estimates of the normalized autocovariance and partial autocorrelation functions should be essentially zero. The Q_L statistics of eq. (18) have chi square distributions in the cases of an ARIMA model, transfer function model alone, or a transfer function plus noise model in which no autoregressive

coefficients in the transfer function have been estimated [40,52]. Since the transfer function and noise model parameter estimates are not independent in the more general case, the covariance matrix of the residual normalized autocovariance function is not a diagonal one so that calculation of a chi square variable from the residual autocorrelations is much more complicated than in eq. (18). However, the Q_L statistics can be used as informal indicators of residual whiteness in this case.

In parametric models with a transfer function component, the residual series should be independent of the input. To check residual independence from the input, normalized cross-covariances between these two series are calculated. However, if the input series is correlated, then the normalized cross-covariances between the input and the residuals of a perfectly adequate transfer function will show pronounced patterns [28]. Thus, normalized cross-covariances are performed between the residuals and a prewhitened input. These cross-covariances should be close to zero if the residuals contain no information about the input response. In IDENT, prewhitening of the input is accomplished by fitting an all pole model to the input:

$$x(t) = pa_1x(t-1) + \dots + pa_nx(t-n) + \alpha(t) \quad (24)$$

The pa 's are the autoregressive parameters of the model and $\alpha(t)$ is the residual series or the prewhitened input. The pa 's are estimated quickly and recursively with eq. (15). An efficient estimate of the normalized cross-covariance function is provided by Box and Jenkins [28]*:

$$\hat{r}_{e\alpha}(t) = 1/N \left(\sum_{k=1}^{N-t} [\hat{e}(k) - \hat{m}_e] \times [\alpha(k+t) - \hat{m}_\alpha] \right) / SD_e SD_\alpha \quad (25)$$

where:

\hat{m}_α = prewhitened mean estimate

* Again, the term, normalized cross-covariance, is consistent with engineering usage, whereas [28] and [35] refer to the cross-correlation function.

\hat{m}_e = the residual mean estimate

SD_e = the standard deviation of the residuals

SD_α = the standard deviation of the prewhitened input.

In the situation where no cross-correlation exists, these normalized cross-covariance estimates have an approximate standard deviation of $(N - t)^{-1/2}$ [28]. In IDENT, statistics similar to those in eq. (18) are calculated for the cross-covariance estimates:

$$S_L = N \sum_{t=0}^L \hat{r}_{e\alpha}^2(t) \quad (26)$$

Again, in the case where the parametric model contains a transfer function alone or a transfer function with no autoregressive parameters combined with a noise model, the S_L statistic has a chi square distribution. In other situations, it can be used as an informal indicator of residual independence from the input.

Additional calculations may serve as aids in the model selection process. The singular points of both the transfer function and noise model are found with the subroutine DPRQD from the IBM Scientific Software Package [53]. Near cancellation or small magnitude of these singular points is an indication that overparametrization has occurred.

Also, the frequency response, gain and phase, of the parametric transfer function is calculated as follows [36]:

$$pf(e^{-j\omega}) = \frac{b_0 + b_1 e^{-j\omega} + \dots + b_m e^{-j\omega m}}{1 + a_1 e^{-j\omega} + \dots + a_n e^{-j\omega n}} \quad (27)$$

Here, gain and phase for 16 values of ω are plotted, going from 0 Hz to 1/8, 1/4, 1/2, or to the full Nyquist frequency. This transfer function may be visually compared to a previously calculated nonparametric one.

2.4. Data preparation

Since the input and output time series are assumed to be stationary, especially for the parameter estimation procedure, it may be necessary to

remove any nonstationarities. An efficient method for eliminating nonstationarities is differencing [28]. Differencing the series once removes nonstationary behavior due to nonuniform constant levels. Differencing twice also removes nonstationary behavior resulting from local linear trends in the data. However, the differencing operation is essentially a high pass filter which increases the amount of high frequency noise and thus decreases the signal to noise ratio of the data, which in turn slows the parameter estimator's convergence rate in the case of a transfer function plus noise model. Thus, detrending of the data may be performed instead. In detrending, a regression line is fitted to the data using standard least squares techniques [54]. The line is then subtracted from the data.

2.5. Additional calculations

Some additional calculations are performed in IDENT for transfer function plus noise models as to the relative amount of energy in the data due to the noise model and to the transfer function component. The signal variance is obtained as follows:

$$\text{Sig} = |pf(e^{-j\omega})|^2 |X(e^{-j\omega})|^2 \quad (28)$$

where:

$pf(e^{-j\omega})$ is calculated as in eq. (27) with ω proceeding in increments of the Nyquist frequency/256

$|X(e^{-j\omega})|^2$ is the input spectrum calculated nonparametrically

The noise spectrum is calculated similarly:

$$\text{Noise} = |C(e^{-j\omega}) / (f(e^{-j\omega})A(e^{-j\omega}))|^2 \text{Var}_e \quad (29)$$

where:

Var_e is the residual variance

Here, ω extends from 0 Hz to the Nyquist frequency in 256 steps. The noise variance is obtained by multiplying the noise spectrum by the residual variance. The signal to noise ratio is very simply:

$$\text{SNR} = \text{SIG}/\text{NOISE} \quad (30)$$

3. PROGRAM DESCRIPTION

The program processes 11 possible commands, described in order below.

3.1. Identification of the output array

Specification of this command results in a univariate analysis of output data. The univariate analysis consists of computation of means (eq. (12)), variance (eq. (13)), normalized autocovariance lags (eq. (14)), and partial autocorrelation lags (eq. (15)). Also calculated are the Durbin Watson statistic of eq. (19), and the Q_L statistics ($L = 10, 20, 35$) of eq. (18). A univariate spectrum is next calculated.

3.2. Parameter estimation

A 2 command results in the estimation of parameters of a specified model structure. The model orders (transfer function poles, transfer function zeros, noise poles, noise zeros) and deadtime factor must be supplied from a command file or in response to the queries:

ENTER ORDERS: TFP, TFZ, NP, NZ

ENTER DEADTIME FACTOR:

Any of these orders may be 0 to result in the estimation of any subset of a transfer function plus noise model. If TFP is 0 and TFZ is -1 , then the parameters of a univariate ARIMA model are estimated.

3.3. Model evaluation

The model evaluation command is the most complicated one, resulting in several calculations. First, the residual calculation of eq. (23) is performed. A univariate analysis, similar to that described for command 1 except for no spectrum, is then performed on the residuals. Then, in the case of a transfer function evaluation, the cross correlations of eq. (25) and the S_L statistics of eq. (26) are also evaluated. Also for a transfer function, the nonparametric input spectrum, the parametric spectrum, signal variance, noise variance, and sig-

nal to noise ratio of eq. (27)–(30) are calculated. Then, the roots of the transfer function and/or noise model numerator and denominator polynomials are found.

3.4. Identification of the input array

This command results in the processing of the input data in the manner described in command 1.

3.5. Identification of the noise array

A prerequisite for this command is the command, 7, so that a nonparametric impulse response is available for the noise series estimation. First, the noise series is estimated according to eq. (22), using a truncated version of the impulse response. A univariate analysis similar to that described for command 1 is performed on the estimated noise series.

3.6. Estimation of prewhitening parameters

In the evaluation of transfer function models, where it is desired to check residual independence from the input, it is necessary to first prewhiten the input. The calculation of the input prewhitening coefficients is done as described in eq. (15). A univariate analysis is done on the resulting residuals in order that the prewhitening model's adequacy may be evaluated. The input residuals are held for later testing of transfer function models.

3.7. Estimation of transfer function and impulse response

In this command, estimation of the nonparametric transfer function is performed according to Fig. 3 and eq. (21).

3.8. Specification of difference parameter or case range

Before any of the other commands can be processed for each set of data, this command must be specified at least once. The case range is specified with a beginning point, NN1, and an endpoint, NN2, which are either from a command file

or typed in response to the query which also specifies the current case range:

THE CURRENT CASE RANGE (1512)

The differencing factors for the input and output data sequences are obtained from the command file or from responses to the queries:

THE CURRENT INPUT DIFFERENCE PARAMETER (0)

THE CURRENT OUTPUT DIFFERENCE PARAMETER (0)

If the operator simply enters a zero in response to any of the above questions, the current case range and differencing parameters in parentheses are retained. In addition, regression lines may be fitted to and subtracted from the input and output data between the endpoints.

3.9. Change operation parameters

There are 8 operation parameters set to their default values in the main program which are listed in Table 1. Any of these operation param-

eters may be changed by typing in the desired value in response to the query. By typing a 0 in response to these queries, the operator leaves the operation parameters unchanged.

3.10. Stop

A 10 command causes the analysis to stop on the current data set. If there is more data, either in the current file, or in other files which were set up at the start of the program, then the new data is read in.

3.11. Help

By typing an 11, a user can get documentation on the use of any of the first 10 commands.

4. SAMPLE RUN

The data used for this run consist of a 512 point input array representing the position of a posture platform and a 512 point output array represent-

TABLE 1
Operation parameters

Name	Allowed values	Default value	Function
NA	Any positive integer up to 50	10	Number of autocorrelation lags printed out in univariate analysis.
NP	Any positive integer up to 50	10	Number of partial autocorrelation lags printed out in univariate analysis.
IMP	Any positive integer up to 50	30	Length of impulse response estimate to be used in estimation of noise series.
LAG	1 or 2	1	The lag window size for spectral smoothing is 2^{-LAG} , either 1/2 or 1/4 of the correlation function length.
KSAMP	1,2,4,8	8	Resampling factor for spectral calculations. The data is resampled in the frequency domain to $KSAMP^{-1}$ of the Nyquist frequency.
NPW	Any positive integer up to 50	17	Number of poles used in prewhitening the input.
ITREND	1 or -1	1	Detrending factor. If ITREND is 1, the data are detrended.
IDEV	3 or 5	5	Command input device. If IDEV = 5, commands are read from the terminal, if IDEV = 3, they are read from a command file.

ing the position of a subject's waist. In fig. 5(a-e) is shown a typical run.

In fig. 5(a), the program is started. A request for the sampling rate is answered by typing the floating point sampling rate in Hz. The input and output data are stored in 512 point arrays in 1615 format in the file PP03C.DAT. In response to the prompt for a command, a 9 is typed so that current values of the operation parameters can be viewed. A message is typed which lists the operation parameters and their appropriate values. Then in response to all prompts for values, the operator types a zero so that the numbers appearing in parentheses after each prompt remain the current values of these operation parameters. Next, an 8 is typed in response to the prompt for a command, so that a case range may be established. The operator's responses indicate that points 1-464 of the input and output arrays are to be used and that the input array is to be differenced. Since the detrending operation parameter is set to 1, the input and output arrays are detrended first. The regression line's slope and intercept and the variance of the detrended (but not differenced) data are printed for each array.

In Fig. 5(a), a 6 command is indicated, so that the differenced input array is prewhitened. The operation parameter for the number of prewhitening poles is set to 17, so that a 17 pole model is fitted to the input data. In the first part of the output, the mean, variance, 10 normalized autocovariance lags, 17 partial autocorrelation lags, a Durbin-Watson statistic, and Q statistics along with appropriate chi square statistics at the .05 and .10 levels are printed out. It can be seen that 17 poles are enough to whiten the differenced input since the last few partial autocorrelation lags are on the order of their approximate standard deviations, $464^{-1/2}$. The next part of the output illustrates the diagnostics applied to the residuals of the 17 pole model (the prewhitened input). Again there appear the univariate statistics. Since the Durbin-Watson statistic is close to 2.0 and the Q_{35} statistics do not exceed their .10 probability levels, one can assume that the differenced input series has been adequately whitened for future use in testing transfer function plus noise models. The 17 poles of the model are also listed.

The next command illustrated in Fig. 5(b) is an 8. The case range is changed to 209-464 and no differencing of the input is done. A 7 command results in an estimate of the resampled parametric transfer function and impulse response. Sixteen values up to 1.56250 Hz of the transfer function's magnitude follow by sixteen values of its phase in radians are shown. This transfer function with a magnitude peak near 0 frequency suggests that perhaps one pole is sufficient to model the transfer function.

In Fig. 5(c), a command of 8 enables the case range to be changed again. This time, the first endpoint is set to 22 since the prewhitened input is not defined for values earlier than that. Next, a command of 2 results in the estimation of a parametric transfer function plus noise model. In this case, 1 pole and 0 zeros for the transfer function and 2 zeros and 1 pole for the noise model are specified. A deadtime factor of 4 is also indicated. Printed are the parameter estimates at the end of each iteration through the data. These parameter estimates are shown in the following order: transfer function numerator then denominator, noise model numerator then denominator. When all these estimates change by less than .01, the estimation stops.

Lastly, Fig. 5(d) illustrates the diagnostics applied to the residuals of the above model. It can be seen that the residual mean is close to 0, and that the estimates of the normalized autocovariance lags and partial autocorrelation lags are small. Also, the Q statistics are not chi square variables in this case, but they are too small to arouse serious concern about model adequacy. Similarly, estimates of lags of the normalized crosscovariance function between the residuals and prewhitened input are small as indicated both by the first 11 values individually and by the 3 S statistics.

Next are printed the noise variance, signal variance, and signal to noise ratios as well as plots of the parametric transfer function gain and phase lags up to 1.5625 Hz. When the effect of the deadtime is considered, this transfer function seems to be a reasonably smooth approximation to the nonparametric one illustrated in Fig. 5(c). Also, it can be seen from the list of the parametric model's poles and zeros that all of the singular points seem

R IDENT
 ENTER THE SAMPLING RATE (HZ): 25.
 ENTER THE NUMBER OF POINTS IN A
 SINGLE DATA ARRAY 512
 ENTER THE NAMES OF FILES (UP TO 6) OF DATA TO BE
 ANALYZED, SEPARATED BY COMMAS.
 *FF03C.DAT

ENTER THE FORMAT OF THE DATA VARIABLES.
 EXAMPLE: (1615) (1615)

***** CAMERA 1 *****

COMMAND:9

LIST OF OPERATION PARAMETERS

NO. AUTOCORRELATION LAGS (UP TO 50)
 NO. PARTIAL AUTOCORRELATION LAGS (UP TO 50)
 LAG WINDOW FACTOR = 1/2 TO THE FACTOR
 RESAMPLING FACTOR = 1,2,4,8
 1 IS BEST FOR OBTAINING A NOISE ESTIMATE.
 IMPULSE RESPONSE LENGTH FOR ESTIMATING NOISE (UP TO 50)
 NO. PREWHITENING POLES (UP TO 50)
 DETRENDING FACTOR = 1 FOR DETRENDING, -1 FOR NOT
 COMMAND INPUT DEVICE = 5 FOR TERMINAL, 3 FOR FILE

NUMBER OF AUTOCORRELATIONS(10)0
 NUMBER OF PARTIAL AUTOCORRELATIONS(10)0
 THE LAG WINDOW(1)0
 THE IMPULSE RESPONSE LENGTH(30)0
 THE CURRENT RESAMPLING FACTOR(8)0
 THE NUMBER OF PREWHITENING POLES(17)0
 THE DETRENDING FACTOR(1)0
 THE COMMAND INPUT DEVICE(5)0

COMMAND:8

THE CURRENT CASE RANGE(0 0)1,464
 THE CURRENT INPUT DIFFERENCE PARAMETER(0)1
 THE CURRENT OUTPUT DIFFERENCE PARAMETER(0)0

SLOPE INTERCEPT SIGNAL VARIANCE

INPUT 0.01889 -4.39109 1.03621

OUTPUT 0.01513 -3.52503 3.70571

COMMAND:6

*****ESTIMATION OF PREWHITENING PARAMETERS*****

MEAN= 0.00421 VARIANCE= 0.05259

-AUTOCORRELATIONS-

LAG	VALUE	-1	0	+1
1	-0.1674		*	
2	0.2171			*
3	0.1003			*
4	0.0908			*
5	-0.0740		*	
6	0.0213		*	
7	-0.0905		*	
8	-0.0229		*	
9	-0.0910		*	
10	0.0333		*	

-PARTIAL AUTOCORRELATIONS-

LAG	VALUE	-1	0	+1
1	-0.1674		*	
2	0.1945			*
3	0.1734			*
4	0.0985			*
5	-0.1131		*	
6	-0.0772		*	
7	-0.0972		*	
8	-0.0261		*	
9	-0.0423		*	
10	0.0560		*	
11	0.0100		*	
12	0.0603		*	
13	0.0282		*	
14	0.0387		*	
15	-0.0202		*	
16	-0.0788		*	
17	-0.0389		*	

DURBIN-WATSON STATISTIC = 2.33446

Q10 Q20 Q35

5(a)

DEG. FREED.	10	20	35
.05 LEVELS	18.3	31.4	49.8
.10 LEVELS	16.0	28.4	46.1
ACTUAL	54.4	69.8	86.1

*****DIAGNOSTICS*****

THE CURRENT CASE RANGE = 19 464

-----RESIDUAL STATISTICS-----

MEAN= 0.00240 VARIANCE= 0.04644

-AUTOCORRELATIONS-

LAG	VALUE	-1	0	+1
1	-0.0067		*	
2	-0.0066		*	
3	-0.0015		*	
4	0.0021		*	
5	0.0065		*	
6	0.0101		*	
7	0.0029		*	
8	0.0149		*	
9	0.0028		*	
10	-0.0033		*	

-PARTIAL AUTOCORRELATIONS-

LAG	VALUE	-1	0	+1
1	-0.0067		*	
2	-0.0067		*	
3	-0.0016		*	
4	0.0020		*	
5	0.0065		*	
6	0.0102		*	
7	0.0032		*	
8	0.0151		*	
9	0.0030		*	
10	-0.0031		*	

DURBIN-WATSON STATISTIC = 2.01228

Q10 Q20 Q35

DEG. FREED.	0	3	18
.05 LEVELS	0.0	7.8	28.9
.10 LEVELS	0.0	6.2	26.0
ACTUAL	0.2	5.9	20.8

MAG FREQ(HZ)

TRANSFER FUNCTION ZEROS		
TRANSFER FUNCTION POLES		
NOISE ZEROS		
NOISE POLES		
1	0.91034	1.91003
2	0.91034	-1.91003
3	0.85745	0.69179
4	0.85745	-0.69179
5	0.86664	3.89488
6	0.86664	-3.89488
7	0.86205	5.60470
8	0.86205	-5.60470
9	0.85568	7.20042
10	0.85568	-7.20042
11	0.83062	8.57293
12	0.83062	-8.57293
13	0.81969	10.22164
14	0.81969	-10.22164
15	0.86424	12.50001
16	0.62457	11.23522
17	0.62457	-11.23522

COMMAND:8

THE CURRENT CASE RANGE(3 464)209,464
 THE CURRENT INPUT DIFFERENCE PARAMETER(0)0
 THE CURRENT OUTPUT DIFFERENCE PARAMETER(0)0

SLOPE INTERCEPT SIGNAL VARIANCE

INPUT 0.02674 -8.99699 0.90680

OUTPUT 0.02208 -7.43126 4.37324

COMMAND:7

EST. OF TRANSFER FUNCTION AND IMPULSE RESPONSE

GAIN

5(b)

J DENSITY
 0 2.6234 *****
 1 2.6143 *****
 2 2.1645 *****
 3 1.2648 *****
 4 0.4543 **
 5 0.6257 ****
 6 0.3158 *
 7 0.0989 *
 8 0.5084 ***
 9 1.4015 *****
 10 0.3055 *
 11 0.5876 ***
 12 0.3697 **
 13 0.3256 **
 14 0.6785 ****
 15 6.0754 *****

1.56250 HZ

PHASE
 J DENSITY
 0 -0.2616 *****
 1 -0.4350 *****
 2 -0.6881 *****
 3 -1.2675 *****
 4 -2.1697 *****
 5 -2.9484 *****
 6 -3.0726 *****
 7 -1.8480 *****
 8 -1.3663 *****
 9 -1.5872 *****
 10 -2.0957 ****
 11 -2.5344 *
 12 -2.3482 *
 13 -1.3601 *****
 14 -1.6352 *****
 15 -0.8362 *****

1.56250 HZ

IMPULSE RESPONSE
 J DENSITY
 0 0.2944 *****
 1 0.5897 *****
 2 0.5115 *****
 3 0.3463 *****
 4 0.0552 *****
 5 0.1507 *****
 6 -0.1008 *
 7 -0.0583 ***
 8 -0.0137 *****
 9 -0.1177 *
 10 -0.0453 ****
 11 0.0925 *****
 12 0.0512 *****
 13 0.0231 *****
 14 0.0086 *****
 15 0.0140 *****
 16 -0.0629 ***
 17 0.0036 *****
 18 -0.0006 *****
 19 0.0043 *****
 20 0.0213 *****
 21 0.0110 *****
 22 -0.0035 *****
 23 -0.0167 *****
 24 0.0118 *****
 25 -0.0121 *****
 26 0.0029 *****
 27 0.0065 *****
 28 0.0001 *****
 29 0.0046 *****
 30 -0.0118 *****
 31 0.0170 *****
 32 -0.0237 *****
 33 0.0262 *****
 34 -0.0352 *****
 35 0.0376 *****
 36 -0.0330 *****
 37 0.0392 *****
 38 -0.0511 *****
 39 0.0566 *****

COMMAND: B
 THE CURRENT CASE RANGE(209 464)22,464
 THE CURRENT INPUT DIFFERENCE PARAMETER(0)0
 THE CURRENT OUTPUT DIFFERENCE PARAMETER(0)0

	SLOPE	INTERCEPT	SIGNAL VARIANCE
INPUT	0.01826	-4.43774	1.03305
OUTPUT	0.01345	-3.26870	3.48634

COMMAND: D

ENTER ORDERS: TFF,TFZ,NF,NZ 1,0,1,2
 ENTER DELAY FACTOR4
 THE DEADTIME FACTOR = 4

NUMBER OF PARAMETERS:TFN: 1 TFD: 0 NN: 2 ND: 1

ITERATION	PARAMETERS			
0	0.10986	0.92570	-1.35662	0.48424 0.93130
1	0.11963	0.93598	-1.44533	0.55648 0.94314
2	0.13098	0.92279	-1.45676	0.57074 0.95137
3	0.13930	0.91393	-1.46128	0.57693 0.95571

ESTIMATED STANDARD DEVIATIONS:
 STD DEV. 0.02511 0.02791 0.04670 0.04336 0.02201

THE RECURSIVELY ESTIMATED RESIDUAL VARIANCE= 0.12195

COMMAND: 3

*****DIAGNOSTICS*****

THE CURRENT CASE RANGE = 23 460

-----RESIDUAL STATISTICS-----
 MEAN= 0.00631 VARIANCE= 0.11137

-AUTOCORRELATIONS-

LAG	VALUE	-1	0	+1
1	-0.0320		*	
2	-0.0154		*	
3	0.0305		*	
4	0.0210		*	
5	-0.0701		*	
6	-0.0245		*	
7	0.0319		*	
8	-0.0220		*	
9	0.0112		*	
10	0.0795		*	

-PARTIAL AUTOCORRELATIONS-

LAG	VALUE	-1	0	+1
1	-0.0320		*	
2	-0.0165		*	
3	0.0295		*	
4	0.0228		*	
5	-0.0679		*	
6	-0.0294		*	
7	0.0271		*	
8	-0.0172		*	
9	0.0152		*	
10	0.0751		*	

DURBIN-WATSON STATISTIC = 2.06380

	Q10	Q20	Q35
DEG. FREED.	7	17	32
.05 LEVELS	14.1	27.6	46.2
.10 LEVELS	12.0	24.8	42.6
ACTUAL	7.0	12.3	22.4

-----S TEST-----
 MEAN= 0.00631 VARIANCE= 0.11137
 MEAN= 0.00402 VARIANCE= 0.04617

THE CROSS- CORRELATION COEFFICIENT = 0.086525

-CROSS CORRELATIONS-

LAG	VALUE	-1	0	+1
1	0.0510		*	
2	0.0427		*	
3	0.0171		*	
4	-0.0868		*	
5	0.0644		*	
6	-0.0163		*	
7	-0.0049		*	
8	0.0245		*	
9	0.0377		*	
10	-0.0505		*	

DURBIN-WATSON STATISTIC = 2.06380

	Q10	Q20	Q35
DEG. FREED.	9	19	34
.05 LEVELS	16.9	30.1	48.6

*****MODEL PARAMETER ESTIMATION***** 5(c)

5(d)

```

.10 LEVELS 14.7      27.2      44.9
ACTUAL     12.7      28.4      40.1

N= 1.8138311 S= 2.1531432 SNR= 1.1870693

PARAMETRIC TRANSFER FUNCTION SPECTRUM

GAIN
J : DENSITY
-----+-----
 0  1.6037 *****
 1  1.4982 *****
 2  1.3375 *****
 3  1.1712 *****
 4  1.0229 *****
 5  0.8985 *****
 6  0.7561 *****
 7  0.7119 *****
 8  0.6423 *****
 9  0.5841 *****
10  0.5351 *****
11  0.4933 *****
12  0.4574 *****
13  0.4262 *****
14  0.3989 *****
15  0.3749 *****

1.56250 HZ

PHASE
J : DENSITY
-----+-----
 0  -0.1295 *****
 1  -0.3702 *****
 2  -0.5682 *****
 3  -0.7198 *****
 4  -0.8327 *****
 5  -0.9163 *****
 6  -0.9786 *****
 7  -1.0254 *****
 8  -1.0608 *****
 9  -1.0876 *****
10  -1.1079 *****
11  -1.1231 *****
12  -1.1344 *****
13  -1.1424 *****
14  -1.1479 *****
15  -1.1511 *****

1.56250 HZ

MAG          FREQ(HZ)
TRANSFER FUNCTION ZEROS
TRANSFER FUNCTION POLES
 1  0.91393      0.00000
NOISE ZEROS
 1  0.75956      1.10147
 2  0.75956     -1.10147
NOISE POLES
 1  -1.95571     0.00000

COMMAND:

```

Fig. 5. Sample run.

to be significant. Thus, from examination of all the model evaluation criteria available here, one could conclude that an acceptable model has been found. However, it is probable that this model is not a unique one.

5. HARDWARE AND SOFTWARE SPECIFICATIONS

The program is designed to run on a LSI-11 minicomputer with the minimum hardware/software components:

- (1) 28 K (16 bits words) of memory.
- (2) DECWRITER III hard copy terminal.
- (3) Dual floppy disk.
- (4) RT-11 Version IV Operating System.

All parts of the program are written in FORTRAN IV.

All computations except for those of the root finding algorithm are done in single precision arithmetic.

The decimal core requirements of the driver and each overlay are as follows:

- (1) Root segment including main program and driver subroutine occupies 14881 words.
- (2) PAREST overlay occupies 5550 words.
- (3) The UNIVRT overlay occupies 5546 words.
- (4) The SING overlay occupies 6154 words.
- (5) The NONPAR overlay occupies 8534 words.
- (6) The RSD overlay occupies 7436 words.
- (7) The TEACH overlay occupies 8655 words.

6. MODE OF AVAILABILITY

Source program listings, a users manual, a programming guide, and reprints may be obtained from the second author upon request at the following address: Dr David J. Anderson, Kresge Hearing Research Institute, University of Michigan Medical School, 1301 E. Ann St., Ann Arbor, MI 48109, USA.

REFERENCES

- [1] G.A. Bekey, Parameter estimation in biological systems: a survey, *in*: Proc. 3rd IFAC Symp. Identification and System Parameter Estimation, The Hague/Delft, ed. P. Eykhoff (Elsevier/North-Holland, Amsterdam, New York, NY, 1973).
- [2] G.A. Bekey and J.E.W. Beneken, Identification of biological and medical systems, *in*: Proc. 4th IFAC Symp. Identification and System Parameter Estimation, Tbilisi, ed. N.S. Rajbman (Elsevier/North-Holland, Amsterdam, New York NY, 1978).

5(e)

- [3] E.R. Carson, C. Cobelli and L. Finkelstein, The identification of metabolic systems – a review, *in: Proc. 5th IFAC Symp. Identification and System Parameter Estimation*, Darmstadt, ed. R. Isermann (Pergamon, New York NY, 1980).
- [4] R.E. Poppele, Systems approach to the study of muscle spindles, *in: Control of Posture and Locomotion*, ed. R.B. Stein et al. (Plenum, New York NY, 1973).
- [5] L. Stark, Neurological control systems, studies in bioengineering (Plenum, New York NY, 1968).
- [6] G. Gantchev and V. Popov, *Agressologie* 14/C (1973) 91.
- [7] G.C. Joyce and P.M.H. Rack, *J. Physiol.* 240 (1974) 375.
- [8] P.B.C. Matthews and R.B. Muir, *Physiology* 302 (1980) 427.
- [9] D.O. Walter, *Exp. Neurol.* 8 (1963) 155.
- [10] L.E. Johnson, *Crit. Rev. Bioeng.* 2 (1974) 1.
- [11] B.K. Berkstresser, S. El-Asfour, J.R. McConnell and B.C. McInnes, *Biosciences* 44 (1979) 157.
- [12] D.M. Wiberg, J.W. Bellville, O. Brovko, R. Maine and T.C. Tai, *IEEE Trans. Auto. Cont.* 24 (1979) 716.
- [13] D. Twieg and L. Nardizzi, *Math. Biosci.* 37 (1977) 81.
- [14] J.J. DiStefano, III, K.C. Wilson, M. Jang and P.H. Mak, *Automatica* 11 (1975) 149.
- [15] D.W. Repperger and A.M. Junker, A study of identification methods and structural modeling techniques on empirical data from a motion study, *in: Proc. Int. Conf. Cybernetics and Society (IEEE, New York NY, 1977)*.
- [16] C.M. Metzler, *J. Am. Stat. Assoc.* 66 (1971) 49.
- [17] H.P. Schmid and G.A. Bekey, Tactile information processing by human operators in control systems, *in: Proc. 1977 Joint Automatic Control Conference (IEEE, New York NY, 1977)*.
- [18] D. Graupe and W.K. Cline, *IEEE Trans. Syst. Man Cybernet.* 5 (1975) 252.
- [19] S.B. Davis, *J. Acoust. Soc. Am.* 58 (1975) S111.
- [20] W. Gersch and J. Yonemoto, Automatic classification of EEG's: a parametric model, new features for classification approach, *in: Proc. 1977 Joint Automatic Control Conference (IEEE) (IEEE, New York NY, 1977)*.
- [21] L.H. Zetterberg, *Math. Biosci.* 5 (1969) 227.
- [22] W. Gersch, *Math. Biosci.* 7 (1970) 205.
- [23] H.H.L. Kwok, *IEEE Trans. Bioeng.* 26 (1979) 405.
- [24] K.J. Astrom and P. Eykhoff, *Automatica* 7 (1971) 123.
- [25] G.C. Goodwin and R.L. Payne, *Dynamic System Identification: Experiment Design and Data Analysis* (Academic Press, New York NY, 1977).
- [26] C.D. Marsden, *Prog. Clin. Neurophysiol.* 5 (1978) 1.
- [27] V. Panuska, *IEEE Trans. Auto. Cont.* 25 (1980) 229.
- [28] G.E.P. Box and G.M. Jenkins, *Time Series Analysis: Forecasting and Control* (Holden-Day, San Francisco CA, 1976).
- [29] L. Ljung, *IEEE Trans. Auto. Cont.* 22 (1977) 551.
- [30] A. Gelb, *Applied Optimal Estimation* (MIT Press, Boston MA, 1974).
- [31] A.H. Jazwinski, *Stochastic Processes and Filtering Theory* (Academic Press, New York, NY, 1970).
- [32] J.L. Talmon and A.J.W. van den Boom, On the estimation of the transfer parameters of process and noise dynamics, *in: Proc. 3rd IFAC Symposium on Identification and System Parameter Estimation*, ed. P. Eykhoff (Elsevier/North-Holland, Amsterdam, New York NY, 1973).
- [33] T. Soderstrom, L. Ljung and I. Gustavsson, Analysis of some on-line identification methods, *in: Proc. 4th IFAC Symp. Identification and System Parameter Estimation*, Tbilisi, ed. N.S. Rajbman (Elsevier/North-Holland, Amsterdam, New York NY, 1978).
- [34] T. Soderstrom, L. Ljung and I. Gustavsson, *Automatica* 14 (1978) 231.
- [35] W.A. Fuller, *Introduction to Statistical Time Series* (Wiley, New York, NY, 1976).
- [36] A.V. Oppenheim and R.W. Schaffer, *Digital Signal Processing* (Prentice-Hall, Englewood Cliff, NJ, 1975).
- [37] J. Durbin, *Rev. Int. Inst. Stat.* 28 (1960) 233.
- [38] M.S. Bartlett, *Suppl. J. Roy. Statist. Soc.* 8 (1946) 27.
- [39] M.H. Quenouille, *J. Roy. Stat. Soc. B11* (1949) 68.
- [40] G.E.P. Box and D.A. Pierce, *J. Amer. Statist. Assoc.* 65 (1970) 1509.
- [41] J. Durbin and G.S. Watson, *Biometrika* 37 (1950) 409.
- [42] J. Durbin and G.S. Watson, *Biometrika* 38 (1951) 159.
- [43] A. Papoulis, *Signal Analysis* (McGraw-Hill, New York NY, 1977).
- [44] J.W. Cooley and J.W. Tukey, *Math. Comput.* 19 (1965) 297.
- [45] G.M. Jenkins and D.G. Watts, *Spectral Analysis and Its Applications* (Holden-Day, San Francisco (A), 1968).
- [46] P. Eykhoff, *System Identification* (Wiley, New York NY, 1974).
- [46] H. Takeda, G. Sato, I. Kimura and S. Kanno, Diagnosis of patients with nervous system diseases by their control characteristics in manual control system, *in: Proc. Int. Conf. Cybernetics and Society (IEEE, New York NY, 1978)*.
- [47] T. Soderstrom, *Int. J. Cont.* 26 (1977) 1.
- [48] H. Unbehauen and B. Gohring, Tests for determining model order in parameter estimation, *in: Proc. 3rd IFAC Symp. Identification and System Parameter Estimation*, ed. P. Eykhoff (Elsevier/North-Holland, Amsterdam, New York NY, 1973).
- [49] A.J.W. van den Boom and A.W.M. van den Enden, The determination of the orders of process and noise dynamics, *in: Proc. 3rd IFAC Symp. Identification and System Parameter Estimation*, ed. P. Eykhoff (Elsevier/North-Holland, Amsterdam, New York NY, 1973).
- [50] C.M. Woodside, *Automatica* 7 (1971) 727.
- [51] P. Newbold, Model checking in time series analysis, presented at ASA-Census-NBER Conf. Applied Time Series Analysis of Economic Data, Washington DC, 13–15 Oct. 1981.
- [52] D.A. Pierce, On model checking in time series analysis, presented at aSA-Census-NBER Conf. Applied Time Series Analysis of Economic Data, Washington DC, 13–15 Oct. 1981.
- [53] Subroutine DPRQD, IBM Scientific Subroutine Package.
- [54] J. Neter and W. Wasserman, *Applied Linear Statistical Models* (R.D. Irwin, Homewood IL, 1974).