An Efficient Approach to ARMA Modeling of Biological Systems with Multiple Inputs and Delays

Michael H. Perrott and Richard J. Cohen,* *Member, IEEE*

***Abstract*—This paper presents a new approach to AutoRegressive Moving Average (ARMA or ARX) modeling which automatically seeks the best model order to represent investigated linear, time invariant systems using their input/output data. The algorithm seeks the ARMA parameterization which accounts for variability in the output of the system due to input activity and contains the fewest number of parameters required to do so. The unique characteristics of the proposed system identification algorithm are its simplicity and efficiency in handling systems with *delays* and *multiple inputs*. We present results of applying the algorithm to simulated data and experimental biological data. In addition, a technique for assessing the error associated with the impulse responses calculated from estimated ARMA parameterizations is presented. The mapping from ARMA coefficients to impulse response estimates is nonlinear, which complicates any effort to construct confidence bounds for the obtained impulse responses. Here a method for obtaining a *linearization* of this mapping is derived, which leads to a simple procedure to approximate the confidence bounds.**

I. Introduction

GIVEN a mathematical description of a physical system, one can analyze its overall behavior and predict the response of its output to different inputs. The difficulty, however, lies in determining such a mathematical description. Indeed, although one can derive a model for some simple systems (such as electric motors) using Newton's laws, there are many systems whose outright complexity makes such a task impossible. In such cases, one has to resort to the numerical techniques of system identification [1].

In attempting to determine empirically the mathematical descriptions associated with linear, time invariant (LTI) systems [2], the AutoRegressive Moving Average (ARMA) model, also referred to as the ARX model [1], is a highly flexible structure that can be used to parameterize the dynamics of such systems. Applied in a system identification context, the ARMA model has been used extensively to represent the dynamics of a wide range of systems that involve everything from brain potentials [3] to the wingtips of sailplanes [4].

Manuscript received August 16, 1993; revised August 25, 1995. This work was supported by NASA under Grant NAGW-3927 and by NIH under Grant 5-R01-HL39291. *Asterisk indicates corresponding author.*

M. H. Perrott is with Harvard-Massachusetts Institute of Technology, Division of Health Sciences and Technology, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: perrott@mit.edu).

*R. J. Cohen is with Harvard-Massachusetts Institute of Technology, Division of Health Sciences and Technology, Building E25, Rm 330D, Massachusetts Institute of Technology, 77 Massachusetts Avenue Cambridge, MA 02139 USA.

Publisher Item Identifier S 0018-9294(96)00374-6.

To illustrate the basic issues associated with ARMA modeling, consider a two input LTI system whose inputs are x_1 and x_2, and whose output is y. The corresponding ARMA model is denoted in difference equation format as follows:

$$y[n] = -\sum_{i=1}^{L} a_i y[n-i] + \sum_{j=s_1}^{f_1} b_{1j} x_1[n-j] + \sum_{k=s_2}^{f_2} b_{2k} x_2[n-k] + e[n]. \quad (1)$$

In this equation, the 'a' coefficients correspond to the autoregressive (AR) parameters, while the 'b' coefficients are moving average (MA) parameters. The sequence $e[n]$ is assumed to be white, Gaussian noise [5] with variance $E(e[n]^2) = \sigma^2$. Note that the inputs (i.e., x_1 and x_2) are assumed to be known, which distinguishes the estimation procedure we will address from that of the ARMA modeling examined in Spectral Analysis literature such as [6, ch. 16] , where the input is assumed to be unknown.

In attempting to represent the given 'physical' system with such a model, two fundamental issues arise

1) Estimation of the model order [i.e., the values of $\{L, s_1, f_1, s_2, f_2\}$ in (1)]
2) Estimation of the parameter values once the model order is known [i.e., the values of the 'a' and 'b' coefficients in (1)].

Fortunately, item two in the above list is straightforward—once the ARMA model order is known, the least squares technique [7] can be used to estimate the ARMA parameters. However, determination of the model order is much more difficult. If the model is overparameterized (i.e., has higher order than necessary), than the impulse response estimates calculated from the resulting ARMA parameters will be overly sensitive to corruption by noise. An underparameterized model will lack the degrees of freedom necessary to represent the dynamics of the corresponding physical system.

Several different algorithms have been published in the last few years that attempt to identify the 'best' ARMA model order of an investigated physical system. Korenberg and Paarmann presented a technique to build up an ARMA model describing a corresponding physical system by adding MA and AR parameters in a pairwise fashion until a mean squared error criterion is satisfied [8]. The operation was made efficient through a procedure based on Gram–Schmidt

orthogonalization. Batill and Hollkamp developed a two stage algorithm to estimate AR and MA parameters separately using free-response and forced response data, respectively [9]. The same researchers also described a technique in [4], which they refer to as the reduced backward method, that attempts to reduce the order of an initially overspecified ARMA model by writing it backward in time. Wahlberg proposed to reduce the order of high-order ARX estimates by means of model reduction via a frequency weighted balanced realization in [10] (his definition of the ARX model and our definition of the ARMA model are equivalent). Finally, Liang *et al.* described an efficient ARMA model order estimation procedure based on evaluation of the MDL criterion using the eigenvalues of the parameter covariance matrix [11].

Unfortunately, none of the above techniques consider systems with multiple inputs or delays [The simplest form of delay amounts to having s_1 or $s_2 \neq 0$ in (1).] In fact, to our knowledge, no automatic procedure exists to identify the ARMA model order associated with such systems. However, an enormous number of physical systems have delays associated with them, from biological [12] to electrothermal systems [1].

To deal with delays, Ljung has suggested an interactive procedure in [1] that relies on the user to gradually infer the correct order using the estimation results of previously guessed models. Unfortunately, such an interactive procedure can be very cumbersome and time consuming, especially when dealing with large numbers of data sets. However, Ljung gives useful practical advice in [13] to the user who must deal with systems containing delays; namely, incorrectly estimated delays are visible in calculated ARMA models. In essence he claims that if s_1 and/or s_2 are chosen to be too small in an estimated ARMA model, resulting in underestimated delays, then the leading 'b' coefficient estimates are small compared to the standard deviation of the error associated with them.

A new algorithm will be presented in this paper that *automatically* estimates the ARMA model order associated with systems that are allowed to have *multiple inputs* and *delays*. The underlying approach of the algorithm is much in the spirit of Ljung's practical advice — the results of an overparameterized model are used to infer the order of the best ARMA parameterization that describes the system. In terms of performance, the algorithm is simple and fast. Results obtained with simulated and experimental data appear extremely promising.

The basic requirements of the presented algorithm are that input and output data sequences associated with the unknown system be provided, and that the user choose an initial overparameterized ARMA model. In order for the algorithm to perform well, the input data must be 'persistently exciting' [1] of order greater than or equal to the order of the initial, overparameterized model — this condition is satisfied if the inputs are independent and white. In practice, the condition of persistence of excitation will often be satisfied if the inputs are independent and broadband.

In addition to the proposed ARMA modeling procedure, a technique is briefly discussed (and derived in Appendix A) to estimate the confidence bounds associated with the impulse response estimates that result from the obtained ARMA estimates. To the authors' knowledge, no procedure currently exists to determine such bounds. However, the impulse response estimates that result from ARMA modeling can provide extremely valuable insights since time domain characteristics such as time constants and time delays of the system are readily observed [12]. It is confusing to examine individual ARMA parameter errors to assess the error in the impulse response — there are often many parameters and their errors affect the impulse response estimates in a nontrivial way. Therefore, we feel that the proposed numerical technique, which provides a quick and intuitive measure of the accuracy of impulse response estimates calculated from an ARMA model, will be an important contribution to the area of biomedical signal processing.

II. Background

A. The ARMA Model

We begin by establishing some basic definitions associated with the Multi Input, Single Output ARMA model. For convenience, we will limit discussion to the two input ARMA model — the extension of results to systems with more inputs should be clear.

Referring to (1), we restate the ARMA model in terms of its Z-transform [2] as follows:

$$A(z)y(z) = B_1(z)x_1(z) + B_2(z)x_2(z) + e(z). \tag{2}$$

in which we have defined the following:

$$B_1(z) \equiv \sum_{k=s_1}^{f_1} b_{1k}z^{-k},$$

$$B_2(z) \equiv \sum_{k=s_2}^{f_2} b_{2k}z^{-k}, \qquad A(z) = 1 + \sum_{k=1}^{L} a_k z^{-k}. \tag{3}$$

From (2), we then define the following input to output transfer function relationships

$$h_1(z) \equiv \frac{y(z)}{x_1(z)} = \frac{B_1(z)}{A(z)} \qquad h_2(z) \equiv \frac{y(z)}{x_2(z)} = \frac{B_2(z)}{A(z)}. \tag{4}$$

Although $e(z)$ is assumed white and Gaussian, the ARMA model can be applied in the presence of nonwhite noise given that certain conditions hold. Specifically, if $e(z) = C(z)/D(z)n(z)$, where $n(z)$ is white, Gaussian noise, then (2) can be modified as

$$\tilde{A}(z)y(z) = \tilde{B}_1(z)x_1(z) + \tilde{B}_2(z)x_2(z) + n(z) \tag{5}$$

where

$$\tilde{A}(z) \equiv \frac{D(z)}{C(z)}A(z)$$

$$\tilde{B}_1(z) \equiv \frac{D(z)}{C(z)}B_1(z)$$

$$\tilde{B}_2(z) \equiv \frac{D(z)}{C(z)}B_2(z). \tag{6}$$

The ARMA model is capable of approximating the above system provided $\tilde{A}(z)$, $\tilde{B}_1(z)$, and $\tilde{B}_2(z)$ can be approximated by finite polynomials in z. This condition is satisfied if $1/C(z)$ is strictly stable [2].

B. Model Comparison

Considerable research has been done regarding the comparison of different model parameterizations [1], which has led to the proposal of several information theoretic criteria. The most popular of such criterion include Akaike's Theoretic Information Criterion (AIC) [14] and Rissanen's Minimum Description Length (MDL) [15]. In regards to the effectiveness of these techniques, the reader is referred to [11], [16]–[20].

The algorithm presented in this paper can be implemented with any criterion that makes use of 'residual error norms' [1], which includes both AIC and MDL. In addition, a hypothesis testing approach can be used. This paper will present results using MDL, with the reader being referred to [21] for a description of the approach taken with hypothesis testing. The results of this paper are easily extended to the AIC criterion.

To use the MDL criterion to evaluate a given model's performance in comparison to other models, one simply obtains an MDL value for the model, MDL(k), as follows:

$$\text{MDL}(k) \equiv \left(1 + k\frac{\log(N)}{N}\right)\frac{\|\varepsilon_k\|^2}{N} \tag{7}$$

which is an approximation given in [13] to the formula presented in [15]. In the above formulation, k is defined to be the number of parameters contained within the model, N the number of samples contained in the input and output data sets, ε_k the prediction error sequence [1], and $\|\cdot\|$ the l_2 norm [22]. Note that $\|\varepsilon_k\|$ is referred to as the 'residual error norm' in this paper. Any given model is said to perform better than all models whose MDL values are greater than its own. Thus, selection of a model using the MDL criterion amounts to finding the model among the candidate set which has the minimum MDL value.

C. Selection of Candidate Model Set

A thorough selection of candidate models to be compared using any of the above criterion leads to a computationally intractable problem when dealing with high order LTI systems containing delays. To explain why the allowance of delays causes so much difficulty, we first mention the standard assumptions made for selecting ARMA models to represent systems without delay, namely

- AR parameters:

$$a_i \neq 0 \text{ for } i \in [1,\ L]; a_i = 0, \text{ elsewhere.}$$

- MA parameters:

$$b_{1j} \neq 0 \text{ for } j \in [0,\ L],$$
$$b_{2k} \neq 0 \text{ for } k \in [0,\ L];\ b_{1j} = b_{2k} = 0, \text{ elsewhere.}$$

Selection of a candidate set of models under the above assumptions follows quite easily — the value of L is simply varied over a range that is deemed to be suitable.

Unfortunately, the above assumptions made on the MA parameters in ARMA modeling break down for systems with delays. In such cases, an appropriate set of constraints for these parameters is

- MA parameters:

$$b_{1j} = 0 \text{ for } j \notin [s_1,\ f_1]$$
$$b_{2k} = 0 \text{ for } k \notin [s_2,\ f_2].$$

The subtlety underlying the above assumption is that we not only lift the previous restriction that $s_1 = s_2 = 0$, but we also allow b_{1j} and b_{2k} to equal zero *within the range* $j \in [s_1, f_1]$ and $k \in [s_2, f_2]$, respectively. Thus, we allow for the nonzero MA parameters to be some *combination* of the total number of MA parameters contained in the sets $j \in [s_1, f_1]$ and $k \in [s_2, f_2]$.

Under the new MA constraint, the user now has to worry about varying 5 parameters, namely $\{L, s_1, s_2, f_1, f_2\}$, as well as the various MA parameter *combinations* possible with whatever values of $\{s_1, s_2, f_1, f_2\}$ are eventually chosen. To illustrate the computational complexity associated with such a search, consider the number of combinations that must be evaluated if we choose some maximum range over which to search. If we set minimum values for s_1 and s_2, and maximum values for f_1,f_2, and L, and then define k_{max} as

$$k_{max} = L_{max} + f_{1max} - s_{1min} + f_{2max} - s_{2min} + 2$$

then the total number of parameter combinations to be searched over to determine which ARMA coefficients are nonzero is approximately $2^{k_{max}}$. If k_{max} were even a modest value such as twenty, this would lead to over a million combinations to be compared.

Strictly speaking, in order to determine the optimal ARMA parameterization associated with a system with delays, all possible parameter combinations must be compared, and the problem becomes intractable for high order systems. However, we will present a simple technique in the remainder of this paper that overcomes the computational complexity through a suboptimal, but very effective, approach. The key to our approach rests in developing an *intelligent* procedure to select candidate parameter combinations.

III. Methods

To explain our proposal for an intelligent model selection procedure, we will first recast the ARMA model into a matrix structure, and examine how least squares forms the parameter estimates. Using the results of this discussion, we will then present a technique by which the estimation results of an initial, overparameterized model are incorporated to produce a set of lower order candidate models. Comparison of the models in this set leads to the selection of a model order to represent the system.

A. The Matrix Format of the ARMA Model

Referring back at the definition of the ARMA model given in (1), it is observed that the AR and MA parameters act on a known set of data (i.e., collected input and output data) to form each output value, with $e[n]$ acting as an unknown noise

source. Collecting the known data, we define a 'data vector' associated with any specified $y[n]$ as follows:

$$\begin{aligned}\boldsymbol{\phi}_t[n]^T = [&x_1[n-s_1] \\ &\cdots x_1[n-f_1] \mid x_2[n-s_2]\cdots x_2[n-f_2] \mid y[n-1] \\ &\cdots y[n-L]].\end{aligned} \tag{8}$$

Similarly, we collect the AR and MA terms into a 'parameter vector'

$$\boldsymbol{\theta}_t^T = [b_{1s_1}\cdots b_{1f_1} \mid b_{2s_2}\cdots b_{2f_2} \mid a_1\cdots a_L]. \tag{9}$$

Note that t corresponds to the dimension of $\boldsymbol{\theta}_t$ (i.e., to the number of nonzero parameters in the ARMA model). Making use of these definitions, we rewrite (1) as

$$y[n] = \boldsymbol{\phi}_t[n]^T\boldsymbol{\theta}_t + e[n]. \tag{10}$$

Assuming that N samples have been collected of the input and output sequences, this data is incorporated into the above equation by incrementing n through the N sample values (i.e., $0 \le n < N$) as follows:

$$\mathbf{y} = \begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[N-1] \end{bmatrix} \quad \boldsymbol{\Phi}_t = \begin{bmatrix} \boldsymbol{\phi}_t[0]^T \\ \boldsymbol{\phi}_t[1]^T \\ \vdots \\ \boldsymbol{\phi}_t[N-1]^T \end{bmatrix}$$

$$\mathbf{e} = \begin{bmatrix} e[0] \\ e[1] \\ \vdots \\ e[N-1] \end{bmatrix}. \tag{11}$$

Using these definitions, the relationship between all of the empirical input, output, and noise data is written for the ARMA model as

$$\mathbf{y} = \boldsymbol{\Phi}_t\boldsymbol{\theta}_t + \mathbf{e}. \tag{12}$$

B. Estimation of the ARMA Model

Suppose that (12) represents the actual or 'true' parameterization of an investigated physical system. (In reality, most physical systems are too complicated to be precisely described by a finite parameter ARMA model, so that $\boldsymbol{\theta}_t$ would be better interpreted as a 'best' parameterization according to some performance criterion.) Since the parameters contained within $\boldsymbol{\theta}_t$, along with their values, are unknown in the context of system identification, it is necessary, in practice, to specify an equation that describes an 'estimation model' selected by the user, namely

$$\mathbf{y} = \boldsymbol{\Phi}_k\hat{\boldsymbol{\theta}}_k + \boldsymbol{\varepsilon}_k. \tag{13}$$

In the above equation, $\hat{\boldsymbol{\theta}}_k$ represents the k ARMA parameter estimates, and $\boldsymbol{\varepsilon}_k$ is the corresponding prediction error sequence.

There are three possible scenarios that may occur when the user selects an estimation model to represent a given system $k > t$, $k < t$, and $k = t$. For the sake of simplicity in notation, we denote $k = t$ to mean that the parameters contained in $\hat{\boldsymbol{\theta}}_k$ are precisely those in $\boldsymbol{\theta}_t$. If k is chosen to be larger than t, there will be 'extra' parameters within $\hat{\boldsymbol{\theta}}_k$ which are defined to be those parameters in the *estimation* model that are not in the *actual* model being sought after. These extra parameters serve only to make the overall estimation model more sensitive to noise. The second scenario, namely $k < t$, leads to an estimation model that does not include all of the true parameters, and therefore lacks the degrees of freedom necessary to represent the system. The last scenario, $k = t$, is obviously the desired choice.

In order to find the estimation model for which $k = t$, we propose an approach that incorporates estimation results from an arbitrary, overly specified model. Thus, we initially choose k as some arbitrary, large number such that $k \ge t$ and define this as the 'maximal model'. A technique will be presented that then attempts to remove the extra parameters from $\boldsymbol{\theta}_k$.

C. The Least Squares Procedure

Given that the initial value of k has been chosen, the parameter estimates $\hat{\boldsymbol{\theta}}_k$ are calculated with the least squares procedure [7]. This operation is described compactly for the ARMA model as

$$\hat{\boldsymbol{\theta}}_k = \left(\boldsymbol{\Phi}_k^T\boldsymbol{\Phi}_k\right)^{-1}\boldsymbol{\Phi}_k^T\mathbf{y}. \tag{14}$$

If we are to compare the chosen maximal model with the true model, it will be necessary to augment the $\boldsymbol{\theta}_t$ vector with the extra parameters contained within $\hat{\boldsymbol{\theta}}_k$. For this purpose, (12) is modified as follows:

$$\mathbf{y} = \boldsymbol{\Phi}_k\boldsymbol{\theta}_k + \mathbf{e} \tag{15}$$

where those parameters in $\boldsymbol{\theta}_k$ that are not in $\boldsymbol{\theta}_t$ have a value of zero by definition.

Combining (14) and (15), we obtain

$$\hat{\boldsymbol{\theta}}_k = \boldsymbol{\theta}_k + \mathbf{v}_k, \quad \text{where } \mathbf{v}_k \equiv \left(\boldsymbol{\Phi}_k^T\boldsymbol{\Phi}_k\right)^{-1}\boldsymbol{\Phi}_k^T\mathbf{e}. \tag{16}$$

The above expression shows that the parameter estimates, $\hat{\boldsymbol{\theta}}_k$, are formed as the addition of the true parameters, $\boldsymbol{\theta}_k$, and a corrupting noise vector $\mathbf{v_k}$, whose autocovariance matrix is

$$E(\mathbf{v}_k\mathbf{v}_k^T) = \sigma^2\left(\boldsymbol{\Phi}_k^T\boldsymbol{\Phi}_k\right)^{-1}. \tag{17}$$

The diagonal elements within the above matrix correspond to the variance, or 'average energy', of the individual elements within $\mathbf{v}_k$, which, in general, varies from parameter to parameter.

D. Defining a S/N Ratio

Our concern lies in determining which parameters within $\hat{\boldsymbol{\theta}}_k$ are true (i.e., have an actual value that is nonzero). Intuitively, the larger in magnitude a parameter estimate is, the less likely it has an actual value of zero. Thus, it seems reasonable to compute the likelihood of each parameter being true by comparing their estimated magnitudes — parameters with large estimated magnitudes are more likely to be true than those with small estimated magnitudes. However, each estimate was shown to be corrupted by noise whose average energy varies from parameter to parameter (16). Therefore, if we are to compare parameter estimates, it is more appropriate to first normalize their values by the level of corruption occurring from noise.

If we think of the true parameter component, $\boldsymbol{\theta}_k$, in each estimate as being the signal, and $\mathbf{v}_k$ as being the noise, then a 'signal to noise (S/N) ratio' can be formed for each individual parameter as follows:

$$S/N(\hat{\boldsymbol{\theta}}_k[i]) \equiv \sqrt{\frac{\boldsymbol{\theta}_k[i]^2}{E(v_k[i]^2)}} \approx \sqrt{\frac{\hat{\boldsymbol{\theta}}_k[i]^2}{E(v_k[i]^2)}} \tag{18}$$

where i designates the i^{th} element of vector $\boldsymbol{\theta}_k$. The rightmost approximation in the above equation is necessary since, in the context of estimation, the actual values of each parameter will be unknown. One can see immediately that the S/N ratio effectively normalizes each estimate by the amount of corruption occurring from noise. Therefore, we propose to use the S/N ratio of each parameter to compare the likelihood of each parameter being 'true'. Parameter estimates with high S/N ratios will be considered more important than those with low S/N ratios. One should note that this approach has much in common with Ljung's hint mentioned in the Introduction — if 'b' coefficient estimates are small compared to the standard deviation of the error associated with them, they are considered to be relatively insignificant.

A point must be made clear regarding the limitations of the S/N ratio — it cannot be used to compare MA and AR terms. Such a comparison would be invalid since it would not be consistent under scaling of the input to output transfer functions. Explaining, it is easy to show via (2) that rescaling of the input to output transfer functions does not affect the parameter *combination* used to describe the system, although the *MA* coefficient *values* will be rescaled appropriately (the AR coefficients remain unchanged). Therefore, a consistent procedure that attempts to identify the correct parameter combination to represent an investigated system should be insensitive to the relative gain of each input to output transfer function. However, since the numerator of the S/N ratios are precisely the parameter values, increasing the gain of a given input to output transfer function will *increase* the S/N ratio of its MA coefficients, but *not change* the S/N ratios of the AR parameters. Thus, comparison of the AR and MA parameters via their S/N ratios would lead to different results under different levels of transfer function gain.

E. A Model Selection Procedure

To develop a consistent 'intelligent' model selection procedure, we are forced to consider AR and MA parameters separately in terms of evaluating their relative likelihood of being true. We therefore present a two step process by which AR parameters are evaluated first, followed by the MA coefficients. This procedure was chosen on the basis that it led to a fast identification procedure and gave excellent results — the limitations of it will be given at the end of this section.

An important point regarding the AR parameters is that they are unaffected by the presence of delays in the system. Thus, while delays cause 'gaps' to appear in the MA coefficients, which necessitates the comparison of *combinations* of these parameters, it is reasonable to hold to the simplifying assumption that no such gaps occur in the AR parameters. (If such gaps do occur in the AR parameters, than the AR order of the system will simply be overestimated by the number of such gaps. In practice, this case is rarely of concern). This translates to assuming that $a_i \neq 0$ for $i \in [1, L]$, implying that it is not necessary to look at combinations of AR parameters and that the value of L is the only item of importance. Following this assumption, we create an initial candidate set of models by simply varying L while retaining all the MA coefficients in the 'maximal model' (i.e., the models in this candidate set contain the same MA coefficients but progressively fewer AR parameters). The MDL criterion is then used to compare the performance of each candidate model, and the best performing one is selected and defined as the 'reduced model'.

Using the 'reduced model' parameterization, the S/N ratios of the MA coefficients are calculated and then used to create another candidate set of models. To form this new set, MA parameters are removed one by one from the reduced model in order of greatest likelihood of being insignificant (i.e., MA parameters with low S/N ratios are removed before those with high S/N ratios). Thus, each model within this new candidate set contains the same AR coefficients but progressively fewer MA parameters. The MDL criterion is again used to compare performance of each candidate model, and that model with best performance is chosen and defined as the 'minimal' model. The minimal model is the algorithm's best guess of the true model of the system.

Pulling together the above facts, we now summarize what we will refer to as the 'Arma Parameter Reduction' (APR) algorithm

1) Select a 'maximal' model — a model that is believed to include all the true parameters of the system.
2) Remove the AR parameters from the maximal model one at a time in order of highest index (i.e., decrease L one increment at a time), creating a set of lower order models.
3) Use an evaluation method (i.e., MDL) to choose a best performing 'reduced' model from among the set given by (2). Note that this step implicitly estimates the AR order.
4) Calculate the S/N ratios of each MA coefficient in the reduced model parameterization.
5) Remove the MA parameters one at a time according to their S/N ratios (low S/N ratios first), creating a set of lower order models.
6) Use an evaluation method (i.e., MDL) to choose the best performing 'minimal' model from the set given by (4). Note that this step estimates the MA order, thereby establishing the overall ARMA order.

As a note on the above procedure, the input data sequences should be prescaled to have the same average energy before any of the above steps are performed. This preliminary operation eliminates problems that otherwise occur when comparing the S/N ratios of MA parameters associated with different inputs.

F. Additional Issues

It is appropriate to question the effectiveness of estimating the order of the AR and MA parameters separately. After all,

proper selection of the AR order is somewhat dependant on proper selection of the MA order, and vice versa. Although this is an issue of concern, it is our contention that the outlined procedure works extremely well in most cases. It is useful, however, to observe the conditions under which the AR parameter model order cannot be estimated separately according to step 3 outlined above, causing the algorithm to yield suboptimal results. Proceeding, we bear in mind the definitions of Φ_k and $\hat{\boldsymbol{\theta}}_k$ given in (8) and (9) as we rewrite (13) as

$$\mathbf{y} = [\mathbf{X} \mid \mathbf{Y}] \begin{bmatrix} \hat{\mathbf{b}} \\ \hline \hat{\mathbf{a}} \end{bmatrix} + \varepsilon. \tag{19}$$

Defining the terms in the above equation, $\mathbf{X}$ represents the columns in Φ_k corresponding to the MA parameter estimates contained in $\hat{\mathbf{b}}$, while $\mathbf{Y}$ represents the columns in Φ_k corresponding to the AR parameter estimates contained in $\hat{\mathbf{a}}$. Now, simply stated, the AR order will be unaffected by the inclusion of extra MA parameters in $\hat{\boldsymbol{\theta}}_k$ during step 3 of the APR algorithm if the column space spanned by the columns in $\mathbf{X}$ associated with the 'extra' MA parameters does not significantly overlap that of the columns in $\mathbf{Y}$ that correspond to the 'true' AR parameters. In other words, if the initial selection of the order of the MA parameters (in step 1) is not so high that the *extra* MA coefficients are able to *compensate* for the model performance lost by removing *true* AR parameters (as judged by MDL in step 3), then the AR order estimation will be unaffected by the inclusion of those extra MA parameters. Otherwise, the AR order will have the tendency to be *underestimated* in step 3.

In practice, it is difficult to analytically determine the highest initial MA order that can be selected without adversely affecting the AR order estimation (it will change from system to system). The maximum number of MA coefficients ever needed would simply span the entire duration of their respective impulse response, which would lead to a MA model (i.e., no AR parameters needed). Thus, to create a need for the AR parameters, the MA parameters should span less than the duration of their respective impulse response.

Extending the above discussion to the selection of the initial maximal model, we recommend the following. Initially, run the APR algorithm with an excessively large number of parameters (i.e., an AR order of 10 or higher will overly describe many systems, and the MA parameters should span what is thought to be the duration of their respective impulse response). The results of the algorithm may not be optimal for such a case, but will reveal a reasonable estimate of the duration of the impulse responses. Using this information, the APR algorithm should be run once more with the MA parameters reselected such that they span less than their respecive impulse response (the AR order can be left large). Since removal of excess parameters will tend to smooth the impulse response shapes (as discussed in the Section IV), a *dramatic* change in the *shape* of any impulse responses obtained from the second run of the algorithm is a strong indication that the associated MA order was chosen too low. In such case, the user should repeat the algorithm with the maximal model appropriately changed. A caveat to the above procedure is that in cases where the variance of $\mathbf{e}$ (the corrupting noise) is small, excessive overdescription of the AR parameters can lead to a poorly conditioned estimation procedure [23]. If that happens, the AR order should be lowered as necessary.

A few final points are made regarding the APR algorithm. First, even in cases where the *optimal* parameterization of the system is not correctly selected, the model obtained by the APR algorithm will still be, in general, a better representation of the corresponding system than if a standard ARMA selection procedure is used. This claim is made on the basis that the APR algorithm chooses the set of models to be compared on an intelligent basis, whereas the means of choosing a set of models to be compared in the general method is, in most cases, arbitrary. Second, the requirements for the algorithm to work properly are minimal. The inputs must be 'persistently exciting' [1], meaning they must be independent and broadband (preferably white), and a maximal model must be selected that includes all the true parameters. The constraint on the inputs is true for any system identification algorithm associated with LTI systems. The selection of a maximal model is inherently done in standard ARMA system identification algorithms since a finite set of models must be chosen by the user for comparison. The advantage of the APR procedure is that only *one* model needs to be selected by the user. Finally, the algorithm can be made extremely efficient (and therefore fast) by making use of the fact that the residual error norm values (which are required to compute MDL values) for the model sets chosen in items (2) and (4) can be calculated recursively, the derivation being provided in Appendix B.

G. Confidence Bounds

Once the 'minimal' model has been chosen by the APR algorithm, it is often desirable to calculate the impulse responses corresponding to the estimated ARMA parameters. To gain an idea of the uncertainty in those responses, it is also desirable to calculate confidence boundaries about the resulting estimates. Unfortunately, while the error associated with the ARMA parameters can be readily calculated via (16), the mapping of this error to the corresponding impulse responses is nonlinear and therefore complicates any effort to construct their confidence bounds.

Rather than seek an approach that *exactly* estimates the impulse response confidence bounds, we derive a method in Appendix A by which these bounds can be approximated through a *linearization* of the mapping between ARMA parameters and their respective impulse responses. Since the analysis is done in Appendix A, we will simply make a few qualitative observations concerning the performance of the approach. Namely, the estimated confidence bounds obtained will be very close to the exact bounds if the corrupting noise has a small variance, in which case there is only a small amount of error associated with the estimated impulse responses (i.e., small perturbations to continuous, nonlinear functions are accurately linearized). Under such circumstances, the confidence bounds about the impulse responses will be very tight and accurate. As the variance of the noise increases,

the approximation degrades in its comparison to the exact bounds. However, despite such deviation, the obtained bounds will qualitatively reveal the fact that a large amount of error is associated with the estimates. Thus, the value of the presented technique is that it allows a quick and intuitive *qualitative* assessment of the accuracy of obtained impulse response estimates from ARMA models.

IV. Results

We begin the Results Section by applying the APR algorithm to simulated data and then calculating the confidence bounds associated with the resulting impulse responses. Following this exercise, we then present performance results of the APR algorithm applied to 100 different input/output data sets associated with the same system. Finally, we present the results of applying the algorithm to experimental respiration, blood pressure, and heart rate data.

A. Application of the APR Algorithm to Simulated Data

Consider an example 'physical system' with delays whose optimal ARMA parameterization is as follows:

- AR parameters:

$$L = 2 : \{a_1,\ a_2\} = \{-1.2,\ 0.35\}$$

- MA parameters:

$$s_1 = 3, f_1 = 3 : \{b_{13}\} = \{-1\}$$
$$s_2 = 1, f_2 = 4 : \{b_{21},\ b_{22},\ b_{23},\ b_{24}\} = \{1,\ 0\ ,\ 0,\ -1.3\}.$$

The corresponding input to output transfer functions for the above ARMA model are written as

$$h_1(z) = \frac{-1z^{-3}}{1 - 1.2z^{-1} + 0.35z^{-2}}$$
$$h_2(z) = \frac{1z^{-1} - 1.3z^{-4}}{1 - 1.2z^{-1} + 0.35z^{-2}}. \tag{20}$$

The selection of the above system was motivated by the results of biological data we have encountered, as can be seen by noting the similarity between Fig. 4 (associated with the above system) and Fig. 6 (associated with experimental data). Thus, the example system is not completely contrived; rather, it is actually similar to systems of practical interest.

To simulate the above system, we generated three uncorrelated, white Gaussian noise sequences of length $N = 200$ corresponding to the inputs $x_1[n]$ and $x_2[n]$, and the noise sequence $e[n]$ using the MATLAB software package. Each input was prescaled to have the same energy. Again using MATLAB, we ran these sequences through a simulation of the above system, producing a $y[n]$ sequence as the result. As a measure of the corruption occurring in the output due to noise, we defined the signal to noise ratio of $\mathbf{y}$ as

$$S/N(\mathbf{y}) \equiv \sqrt{\frac{\|h_1[n] * x_1[n] + h_2[n] * x_2[n]\|^2}{\|h_e[n] * e[n]\|^2}} \tag{21}$$

and scaled the variance of $e[n]$ to a value such that the ratio was 1 in our simulation, which led to $\sigma^2 = 3.11$ (note that $*$ in the above equation represents the convolution operation and $h_e[n]$ is the impulse response of $1/A(z)$).

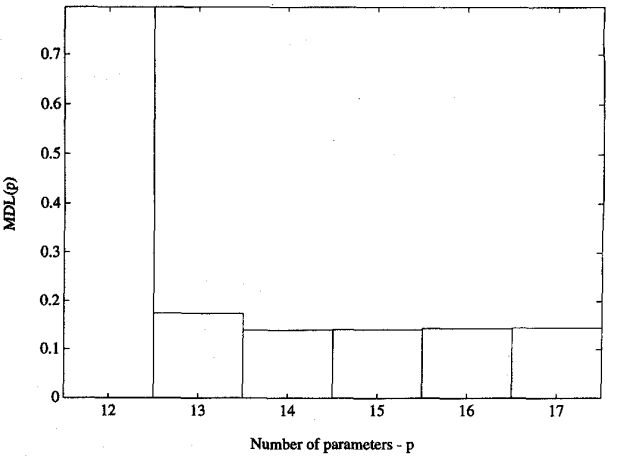


Fig. 1. Plot of MDL criterion applied to the candidate models produced by removal of the AR terms.

Now, beginning with step one of the APR algorithm outline, we first specified a maximal model as

$$\{L, s_1, f_1, s_2, f_2\} = \{5, 0, 5, 0, 5\}. \tag{22}$$

This model was chosen to be only slightly larger than needed to include the true model in order that explanation of the algorithm would be clear and unencumbered. Performance of the algorithm with a larger maximal model will be discussed in the next section.

Proceeding, we removed each of the AR parameters by decreasing L one increment at a time. The MDL values were then calculated for each of the resulting models, producing the bar plot shown in Fig. 1.

Inspection of this plot reveals that the minimum occurs when the number of parameters equals 14 (i.e when 3 of the AR parameters are removed). Thus the reduced model is selected as

$$\{L,\ s_1,\ f_1,\ s_2,\ f_2\} = \{2,\ 0,\ 5,\ 0,\ 5\}. \tag{23}$$

The estimation results of the reduced model were then used to form S/N ratios for each of the MA parameters. A bar plot of these ratios is shown in Fig. 2, inspection of which correctly suggests that b_{13}, b_{21}, and b_{24} are the most likely parameters to be true.

To form a new set of candidate models, we simply proceeded to remove all of the MA parameters, one at a time, from the reduced model in order of least likelihood of being true (i.e., we first removed b_{25}, then b_{22}, then b_{10}, etc.). Following removal of the MA parameters, the two remaining AR terms were also removed (first a_2, then a_1 — this was not necessary, but it does illustrate the effect of removing all the ARMA parameters on the MDL criterion). Thus, fourteen candidate models were created, for which residual error norms were calculated and the MDL criterion used to evaluate the performance of each. Fig. 3 reveals that the model with the minimum MDL value contains 5 parameters (i.e., 3 MA plus 2 AR parameters).

Since b_{13}, b_{21}, and b_{24} have the highest S/N ratios, the 3 MA terms contained in this final selected model correspond

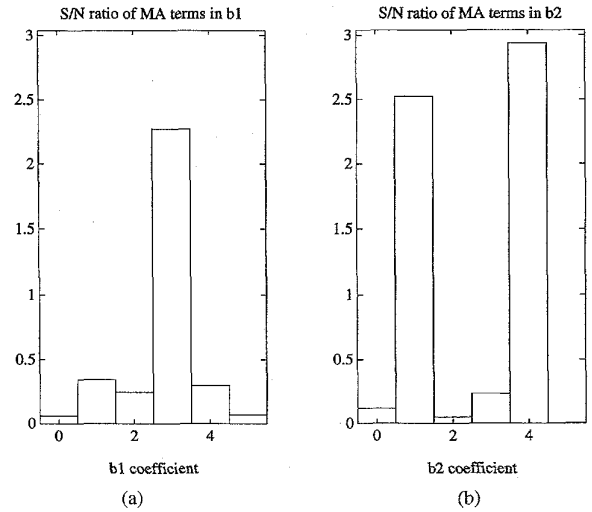


Fig. 2. Plot of the S/N ratios obtained for the MA parameters in the 'reduced model'.

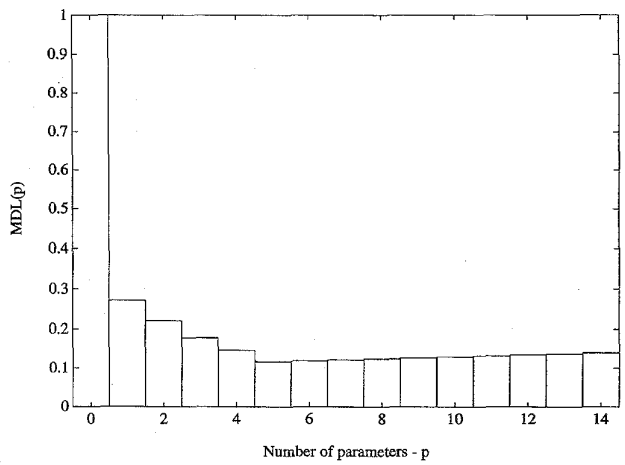


Fig. 3. Plot of MDL criterion applied to the candidate models produced by removal of the MA terms.

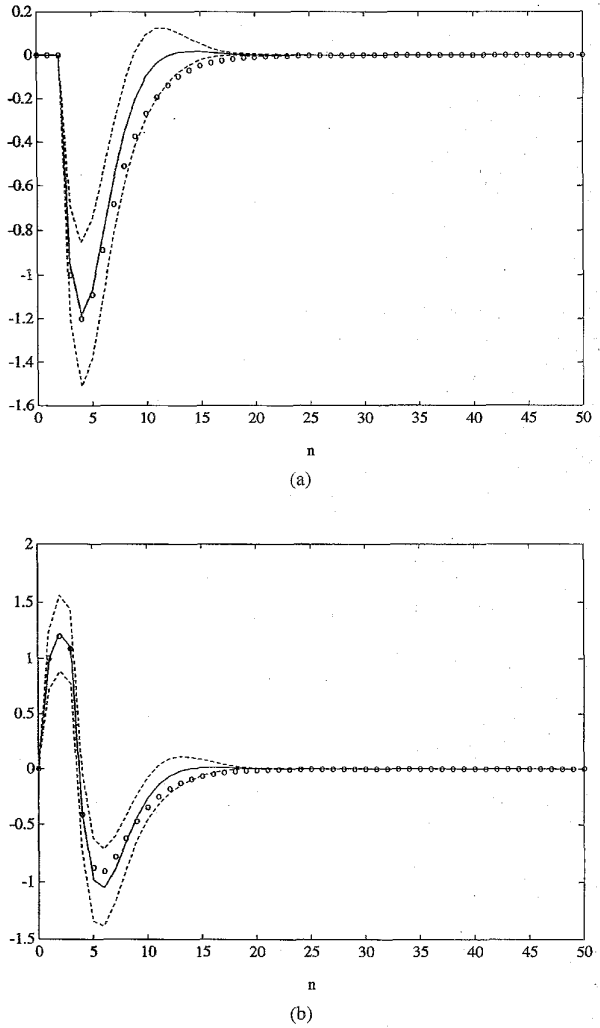


Fig. 4. Plot of the estimated impulse responses (a) $\hat{h_1}[n]$ and (b) $\hat{h_2}[n]$ with their 95% confidence boundary indicated by the dashed lines. The actual impulse responses, (a) $h_1[n]$ and (b) $h_2[n]$, are shown as a series of points.

precisely to these terms. The 'minimal' model order for the system was thus correctly selected as

- AR parameters:

$$L=2:\{a_1,\ a_2\}=\{-1.25,\ 0.43\}$$

- MA parameters:

$$s_1=3, f_1=3:\{b_{13}\}=\{-0.95\}$$
$$s_2=1,\ f_2=4:\{b_{21},\ b_{22},\ b_{23},\ b_{24}\}=\{0.98,\ 0,0,\ -1.26\}$$

Using the above estimated ARMA parameters, we calculated the corresponding impulse responses and then proceeded to determine their 95% confidence bounds using the methodology derived in Appendix A. Referring to the appendix, our first step was to estimate the noise variance via (43), for which we obtained $\hat{\sigma}^2 = 3.29$ (this was very close to the actual value of $\sigma^2 = 3.11$). Following this operation, we applied (42) to produce the plots shown in Fig. 4. One should note that the confidence bounds give a very accurate portrayel of the error associated with the estimated impulse responses.

B. Statistical Evaluation of the APR Algorithm using Simulated Data

To evaluate, in a statistical manner, the performance of the APR algorithm with systems that are *exactly* represented by ARMA models, we applied it to 100 different input/output data sets associated with the example system under 8 different values of $S/N(\mathbf{y})$ and the choice of two different maximal models. As a measure of performance, we first show the success rate at which the APR algorithm correctly identified the ARMA parameterization in the above data. We then show that the average error of the impulse responses calculated from the minimal models selected by the APR algorithm in the above data was significantly lower than the impulse responses associated with the selected maximal models.

In order to obtain the relevant input and output data, a computer was used to generate 100 different data sets of

TABLE I
RATE OF CORRECT MODEL ORDER SELECTION FOR THE EXAMPLE SYSTEM

S/N(y)	correct selection	S/N(y)	correct selection
0.25	91%	0.25	84%
0.5	91%	0.5	87%
0.75	90%	0.75	87%
1.0	90%	1.0	88%
2.0	92%	2.0	84%
3.0	91%	3.0	85%
5.0	93%	5.0	86%
10.0	94%	10.0	85%
maximal model: {5, 0, 5, 0, 5 }		maximal model: {10, 0, 9, 0, 9 }	

input and noise data. Each set consisted of independent, white, Gaussian $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{e}$ sequences of length 1000 samples, and each input sequence was prescaled to have the same energy. Since our desire was to evaluate the performance of the APR algorithm in estimating the optimal model order under different values of $S/N(\mathbf{y})$, its value was varied by appropriately prescaling the variance of $\mathbf{e}$ and then running the resulting sequence, along with the input sequences, through a simulation of the system of Section II.C in order to produce an output sequence. This process was repeated for all 100 data sets for each value of $S/N(\mathbf{y})$, creating a total of 800 data sets.

The success rate of the APR algorithm in correctly estimating the model order of the system in each of the 100 data sets under each $S/N(\mathbf{y})$ value is tabulated in Table I. The leftmost portion of this table indicates the percentage of correct model order estimations (out of 100 data sets) made by the APR algorithm under each $S/N(\mathbf{y})$ ratio with the maximal model selected as $\{5,0,5,0,5\}$. Similarly, the rightmost portion indicates the results obtained with a maximal selected as $\{10,0,9,0,9\}$. The time taken by the APR algorithm to estimate the model order and parameter values associated with a typical data set (1000 samples) when the maximal model was selected as $\{10,0,9,0,9\}$ was 12 seconds on a Sun SPARC station IPX. When the maximal model was set at $\{5,0,5,0,5\}$, that time was reduced to 7 seconds.

There are several important observations to make regarding the above results. First, it should be noted that performance went down when the order of the selected maximal model increased, but not drastically. Thus, in this case, the choice of a severely overparameterized maximal model still yielded acceptable results. Second, the performance was very consistent over a range of $S/N(\mathbf{y})$ ratios. Thus, model order estimation was very robust in this case to changes in $S/N(\mathbf{y})$ and to the choice of a maximal model.

Although it is reassuring that the APR algorithm often *exactly* calculated the correct model order of the simulated system, a more important measure of the performance of the algorithm is the amount of error associated with the impulse responses calculated from the resulting ARMA estimates. To evaluate the performance of the algorithm in this respect, we defined the average impulse response error as

$$\text{average impulse response error} \equiv \frac{1}{M}\sum_{i=1}^{M}\|\hat{\mathbf{h}}_i - \mathbf{h}\|$$

where M corresponds to the number of data sets considered (i.e., $M = 100$ in this case), $\hat{\mathbf{h}}_i$ represents the impulse response estimate associated with data set i, and $\mathbf{h}$ corresponds to the true impulse response of the system.

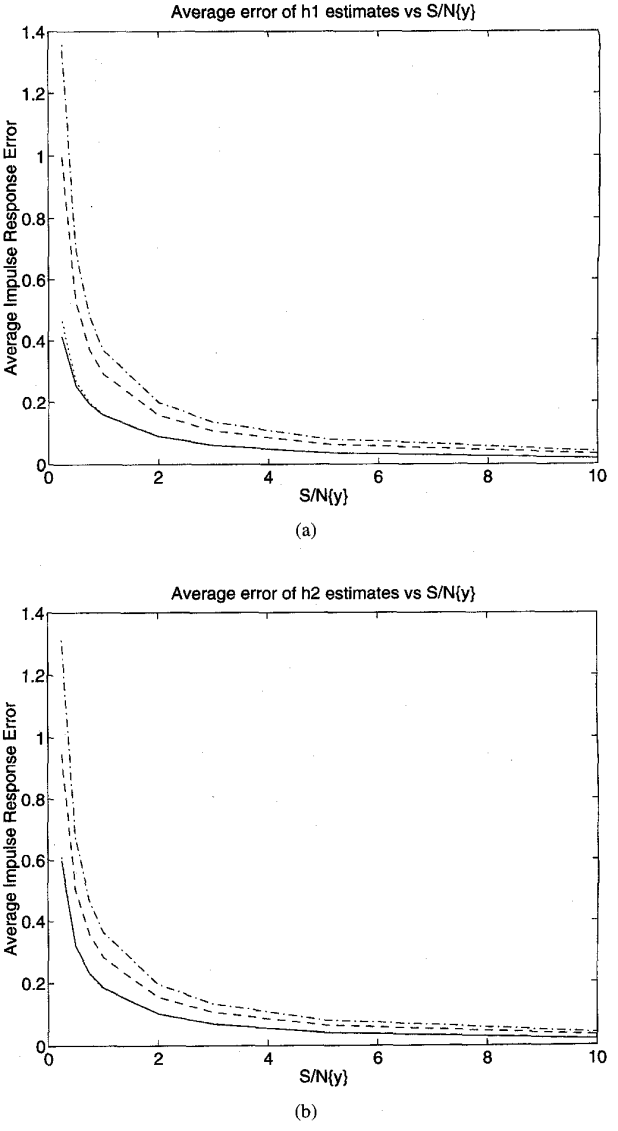


Fig. 5. Comparison of the average impulse response error associated with the maximal model $\{10,0,9,0,9\}$ (dash-dot line), the minimal model under this choice of maximal model occurring after application of the APR algorithm (dotted line), the maximal model $\{5,0,5,0,5\}$ (dashed line), and the minimal model occurring under this last choice of a maximal model (solid line). The top plot (a) shows results corresponding to impulse response h_1; the bottom plot (b) shows results corresponding to impulse response h_2.

Using the above measure of impulse response error, we compared the impulse responses associated with the two maximal models selected above ($\{5,0,5,0,5\}$ and $\{10,0,9,0,9\}$) to the minimal models that resulted after application of the APR algorithm. Using the data considered in forming Table I, Fig. 5 illustrates the error associated with each of the above models under the $S/N(\mathbf{y})$ ratios considered in that table. Glancing at this figure, we first note that the average impulse response error was nearly identical for the minimal models associated with each initial maximal model choice, which supports the fact that the APR algorithm was relatively

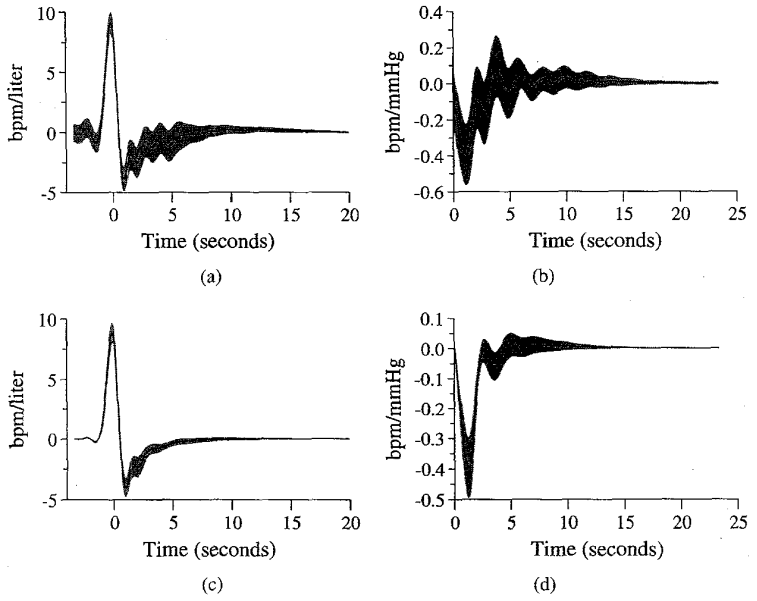


Fig. 6. Impulse response estimates associated with experimental respiration, arterial blood pressure, and heart rate data. (a) Impulse response from ILV to HR for maximal model. (b) Impulse response from ABP to HR for maximal model. (c) Imulse response from ILV to HR for minimal models. (d) Impulse response from ABP to HR for minimal model.

insensitive to the initial choice of a maximal model in this simulation. Further, the average impulse response error of the minimal models selected by the APR algorithm was significantly lower than each of the maximal models under all $S/N(\mathbf{y})$ values. Thus, the algorithm was extremely effective at reducing the amount of error present in impulse response estimates from that of the initially chosen maximal models. In fact, a useful interpretation of the APR algorithm is that it strives to obtain the model order that minimizes the corruption due to noise of the resulting impulse response estimates.

C. Application to Experimental Data

To evaluate the effectiveness of the APR algorithm with systems that can only be *approximated* by ARMA models, we used it to analyze the relationship between fluctuations in instantaneous lung volume (ILV), arterial blood pressure (ABP), and heart rate (HR) data. The relevant data was acquired by Saul *et al.* in an experiment described in [24].

In order to quantify the role of the autonomic nervous system in heart rate variability, the following ARMA relationship has been proposed in [25] as a possible representation of this system's behavior

$$\mathrm{HR}[n] = -\sum_{i=1}^{L} a_i\,\mathrm{HR}[n-i] + \sum_{j=s_1}^{f_1} b_{1j}\,\mathrm{ILV}[n-j] + \sum_{k=s_2}^{f_2} b_{2k}\,\mathrm{ABP}[n-k] + e[n]. \tag{24}$$

In order to estimate the impulse responses associated with the above model, the APR algorithm was applied to ILV, ABP and HR sequences consisting of 1024 data samples at 1.5 Hz.

The maximal model used to represent the above system was chosen through the use of the procedure outlined in the Section III.F as

$$\{L, s_1, f_1, s_2, f_2\} = \{10, -5, 10, 1, 10\}. \tag{25}$$

Note that, by virtue of setting $s_1 = -5$, we have allowed for the possibility that respiration affects heart rate in a noncausal fashion. A brief explanation of this allowance is that the central nervous system controls respiration and may also have a direct influence on HR, so that it is conceivable that heart rate could be affected by respiration activity before it physically occurs [12] [26].

Fig. 6 displays the impulse response estimates for a typical data set before and after the application of the APR algorithm. The top portion of the figure illustrates the responses obtained with the model order set as the maximal model. The impulse responses associated with the minimal model selected by the APR algorithm are shown at the bottom of the figure. The grey region surrounding each response indicates the 95% confidence bound for the estimate, calculated via the method derived in Appendix A. It should be noted that the Matlab function 'interp' was used to obtain a smooth interpolation between the samples of each impulse response estimate.

There are several points to make regarding the impulse response estimates shown in Fig. 6. First, although it is not possible to verify that the minimal model selected was the true model from the results shown, it should be noted that the general shape of the ILV to HR impulse responses correspond well to results published in [12]. In addition, a slightly negative delay from ILV to HR has been reported in previous articles [12] [26], and can be observed in the shown ILV to HR impulse response. As for the performance of the APR algorithm, it should be observed that application of it has the effect of 'smoothing' the impulse response estimates and tightening their confidence bounds with respect to the maximal model estimates.

As a final comment on experimental data, it is tempting to think that the algorithm can be used to find some 'universal' parameterization that will fit a large number of investigated data sets very well. While that is possible in some cases, we have noticed that, in dealing with biological signals, small differences often occur in the impulse responses between data sets which lead to differences in the associated minimal parameterizations (see [12], for instance). Most often, the conveyer of information is the impulse response [12] (or its Fourier transform [2], [24], [26]), not the parameterization. Therefore, the most important characteristic of a good system identification procedure in such cases is that it strive to obtain the model order which minimizes the corruption due to noise of the impulse response estimates. Application of the APR algorithm to the experimental data sets we have encountered, of which the data presented in this paper is typical, show that it does just that — impulse response estimates are 'smoothed' and their confidence bounds tightened.

V. Conclusion

This paper has introduced a simple, efficient procedure for determining the 'best' ARMA model to represent a system using its input and output data. Our approach has been to use the parameter estimates of an overly specified model to construct a set of lower order models that are compared via the MDL criterion. The resulting algorithm is fast (the error norms associated with the candidate models can be computed recursively), requires no user intervention beyond the choice of a maximal model, and easily accommodates systems with multiple inputs and delays.

The algorithm was applied to both simulated and experimental data, and the obtained results were very promising. For a simulated system that was represented exactly by an ARMA model, the algorithm was shown to have a very high success rate in estimating the correct model order. It was also shown to reduce the average error of the resulting impulse responses from that of the initially chosen maximal model. In the case of the experimental data, the result of applying the algorithm was to 'smooth' the impulse response estimates and tighten their confidence bounds with respect to the maximal model estimates. Thus, it was asserted that the APR algorithm strives toward minimizing the corruption due to noise of the ARMA impulse response estimates.

Finally, a technique was discussed (and derived in Appendix A) for the estimation of confidence bounds corresponding to the impulse responses calculated from ARMA parameter estimates. The method is based on obtaining a linearization of the mapping from ARMA estimates to their respective impulse responses. Its value lies in the fact that it provides a quick and intuitive qualitative assessment of the accuracy of the impulse responses calculated from estimated ARMA model parameters.

Appendix A
A Method for Obtaining Impulse Response Confidence Limits

In order to evaluate the effectiveness of an estimation procedure, one must know the uncertainty of the estimates. This section presents a technique for estimating the error associated with the input to output impulse responses obtained from ARMA models.

We begin with the definition of the Z-transform of the impulse response associated with the i^{th} input of an ARMA model

$$h_i(z) = \frac{b_i(z)}{a(z)}. \tag{26}$$

Multiplying both sides of the above equation by $a(z)$, and then inverse Z-transforming the resulting expression, we obtain

$$b_i[n] = h_i[n] * a[n], \tag{27}$$

which states that the MA coefficients associated with any given input are formed by convolving its impulse response with the AR coefficients. This convolution can be written in the form of a matrix multiplication if we consider only the first $k - s_i + 1$ terms of the impulse response

$$\begin{bmatrix} b_i[s_i] \\ b_i[s_i+1] \\ b_i[s_i+2] \\ \vdots \\ b_i[f_i] \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} h_i[s_i] & 0 & 0 & \cdots & 0 \\ h_i[s_i+1] & h_i[s_i] & 0 & & \\ h_i[s_i+2] & h_i[s_i+1] & h_i[s_i] & & \\ \vdots & & & \ddots & \vdots \\ h_i[fi] & & & & \\ \vdots & & & & \\ h_i[k] & & \cdots & & h_i[k-L-1] \end{bmatrix} \cdot \begin{bmatrix} 1 \\ a[1] \\ a[2] \\ \vdots \\ a[L] \end{bmatrix} \tag{28}$$

where we have assumed that k has been chosen to be greater than f_i. As a matter of notation, we will write this matrix multiplication as

$$\begin{bmatrix} \mathbf{b}_i \\ \mathbf{0} \end{bmatrix} = \mathbf{H}_i \begin{bmatrix} 1 \\ \mathbf{a} \end{bmatrix} \tag{29}$$

which implicitly defines the matrix $\mathbf{H}_i$ and the vectors $\mathbf{b}_i$ and $\mathbf{a}$. Alternatively, the given convolution relationship can be written as

$$\begin{bmatrix} b_i[s_i] \\ b_i[s_i+1] \\ b_i[s_i+2] \\ \vdots \\ b_i[f_i] \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ a[1] & 1 & 0 & & \\ \vdots & & \ddots & \ddots & \vdots \\ a[L] & & & & \\ 0 & a[L] & \cdots & & \\ \vdots & & & & \\ 0 & & \cdots & a[1] & 1 \end{bmatrix}$$

$$\cdot \begin{bmatrix} h_i[s_i] \\ h_i[s_i+1] \\ h_i[s_i+2] \\ \vdots \\ h_i[k] \end{bmatrix} \tag{30}$$

which effectively defines the matrix $\mathbf{A}$ and vector $\mathbf{h}_i$ upon rewriting this equation as

$$\begin{bmatrix} \mathbf{b}_i \\ \mathbf{0} \end{bmatrix} = \mathbf{A}\mathbf{h}_i. \tag{31}$$

Let us now examine the effect of perturbing the MA and AR parameters on their associated impulse responses. Defining $\hat{\mathbf{b}}_i$ and $\hat{\mathbf{a}}$ as ARMA estimates, and $\hat{\mathbf{H}}_i$ as the resulting estimate impulse response matrix, we relate the perturbation of each of these estimates via (29) as follows:

$$\begin{bmatrix} \hat{\mathbf{b}}_i \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \Delta\hat{\mathbf{b}}_i \\ \mathbf{0} \end{bmatrix} = \left(\hat{\mathbf{H}}_i + \Delta\hat{\mathbf{H}}_i\right)\left(\begin{bmatrix} 1 \\ \hat{\mathbf{a}} \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta\hat{\mathbf{a}} \end{bmatrix}\right). \tag{32}$$

To explain, the above equation relates the change that occurs in an estimated impulse response ($\Delta\hat{\mathbf{H}}_i$) given that the AR and MA parameters used to calculate it are changed in value ($\Delta\hat{\mathbf{a}}, \Delta\hat{\mathbf{b}}_i$).

Using (29) and the fact that the rightmost portion of (31) and (29) are equal, we rewrite (32) as

$$\hat{\mathbf{A}}\Delta\hat{\mathbf{h}}_i = \begin{bmatrix} \Delta\hat{\mathbf{b}}_i \\ \mathbf{0} \end{bmatrix} + \left(\hat{\mathbf{H}}_i + \Delta\hat{\mathbf{H}}_i\right)\begin{bmatrix} 0 \\ -\Delta\hat{\mathbf{a}} \end{bmatrix}. \tag{33}$$

If we concern ourselves only with small perturbations about $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}_i$, then it is reasonable to assume that the term

$$\Delta\hat{\mathbf{H}}_i\begin{bmatrix} 0 \\ -\Delta\hat{\mathbf{a}} \end{bmatrix}$$

is negligible and therefore can be approximated as zero. Also, the first column of $\hat{\mathbf{H}}_i$ has no effect on the equation since it is multiplied by zero, so that we introduce a new matrix, $\hat{\mathbf{H}}_i^*$, which is defined to be matrix $\hat{\mathbf{H}}_i$ with the first column removed. With these facts in mind, we approximate (33) as

$$\hat{\mathbf{A}}\Delta\hat{\mathbf{h}}_i \approx \begin{bmatrix} \Delta\hat{\mathbf{b}}_i \\ \mathbf{0} \end{bmatrix} + \left(\hat{\mathbf{H}}_i^*\right)[-\Delta\hat{\mathbf{a}}]. \tag{34}$$

A more useful format of the above expression is obtained with slight manipulation

$$\Delta\hat{\mathbf{h}}_i \approx \hat{\mathbf{A}}^{-1}\begin{bmatrix} \mathbf{I}_i & \hat{\mathbf{H}}_i^* \\ \mathbf{0} & \end{bmatrix}\begin{bmatrix} \Delta\hat{\mathbf{b}}_i \\ -\Delta\hat{\mathbf{a}} \end{bmatrix} \tag{35}$$

where we have defined $\mathbf{I}_i$ to be the identity matrix with dimension equal to the number of MA parameters associated with the i^{th} input. Equation 35 effectively provides a *linearization* of the sensitivity of $\hat{\mathbf{h}}_i$ to ARMA parameter changes about the operating point set by their estimates, $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}_i$.

It is straightforward to extend the above results to encompass all of the input to output impulse responses. Assuming M inputs into the system, (35) is expanded as follows:

$$\begin{bmatrix} \Delta\hat{\mathbf{h}}_1 \\ \Delta\hat{\mathbf{h}}_2 \\ \vdots \\ \Delta\hat{\mathbf{h}}_M \end{bmatrix} \approx \begin{bmatrix} \hat{\mathbf{A}}_1^{-1} & 0 & \cdots & 0 & \hat{\mathbf{A}}^{-1}\hat{\mathbf{H}}_1^* \\ 0 & \hat{\mathbf{A}}_2^{-1} & \cdots & 0 & \hat{\mathbf{A}}^{-1}\hat{\mathbf{H}}_2^* \\ 0 & 0 & \ddots & & \vdots \\ 0 & 0 & \cdots & \hat{\mathbf{A}}_M^{-1} & \hat{\mathbf{A}}^{-1}\hat{\mathbf{H}}_M^* \end{bmatrix}$$

$$\cdot \begin{bmatrix} \Delta\hat{\mathbf{b}}_1 \\ \Delta\hat{\mathbf{b}}_2 \\ \vdots \\ \Delta\hat{\mathbf{b}}_M \\ -\Delta\hat{\mathbf{a}} \end{bmatrix} \tag{36}$$

where we have defined

$$\hat{\mathbf{A}}_i^{-1} = \hat{\mathbf{A}}^{-1}\begin{bmatrix} \mathbf{I}_i \\ \mathbf{0} \end{bmatrix}. \tag{37}$$

The rightmost vector of (36) corresponds directly in form to the parameter vector defined in (9). Therefore, it is appropriate to rewrite this expression as

$$\Delta\hat{\mathbf{h}}_{all} \approx \mathbf{D}_k\Delta\hat{\boldsymbol{\theta}}_k \tag{38}$$

where $\Delta\mathbf{h}_{all}$ is a vector containing the resulting change in each of the input to output impulse response vectors due to a perturbation of the estimated parameter vector, $\Delta\hat{\boldsymbol{\theta}}_k$. The matrix $\mathbf{D}_k$ represents a linearization of the true mapping that occurs between these vectors.

Now we are in a position to construct confidence boundaries for the impulse responses. Glancing back at (16), we note that the error associated with the ARMA estimates is expressed as

$$\Delta\hat{\boldsymbol{\theta}}_k \equiv \hat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}_k = \left(\boldsymbol{\Phi}_k^T\boldsymbol{\Phi}_k\right)^{-1}\boldsymbol{\Phi}_k^T\mathbf{e}. \tag{39}$$

Therefore, we can rewrite (38) as

$$\Delta\hat{\mathbf{h}}_{all} \approx \mathbf{D}_k\left(\boldsymbol{\Phi}_k^T\boldsymbol{\Phi}_k\right)^{-1}\boldsymbol{\Phi}_k^T\mathbf{e} \tag{40}$$

and determine confidence bounds for the impulse responses by simply computing the covariance of $\Delta\hat{\mathbf{h}}_{all}$. The covariance matrix is calculated as

$$E\left(\Delta\hat{\mathbf{h}}_{all}\Delta\hat{\mathbf{h}}_{all}^T\right) = \sigma^2\mathbf{D}_k\left(\boldsymbol{\Phi}_k^T\boldsymbol{\Phi}_k\right)^{-1}\mathbf{D}_k^T \tag{41}$$

and the diagonal entries of $E\left(\Delta\hat{\mathbf{h}}_{all}\Delta\hat{\mathbf{h}}_{all}^T\right)$ correspond to the variance associated with each sample of the individual impulse reponse errors.

Note that, since (40) represents a linear mapping, the random variables $\mathbf{h}_{all}[i]$ can be approximated as Gaussian. (The sum of independent Gaussian random variables is also Gaussian.) Therefore, to determine, for instance, error boundaries at 95% confidence for an estimated impulse response sample $\hat{\mathbf{h}}_{all}[i]$, we would calculate

$$\mathbf{h}_{all}[i] = \hat{\mathbf{h}}_{all}[i] \pm 1.96\sqrt{E(\left(\Delta\hat{\mathbf{h}}_{all}\Delta\hat{\mathbf{h}}_{all}^T\right)[i,i])}. \tag{42}$$

As a final comment, it will normally be the case in the context of estimation that σ^2, the variance of the white noise sequence $\mathbf{e}$, is unknown. The following equation provides a straightforward method of estimating this quantity

$$\hat{\sigma}^2 = \frac{1}{N - k_{max}}\|\varepsilon_{k_{max}}\|^2 \tag{43}$$

where N corresponds to the number of data samples in the input and output sequences, k_{max} equals the number of parameters contained within the maximal model selected, and $\|\varepsilon_{k_{max}}\|^2$ is the squared error norm occurring with the maximal model. This last equation is stated in [27] and derived in [21].

APPENDIX B
RECURSIVE COMPUTATION OF ARMA ESTIMATES

The proposed model selection procedure in this paper revolves around the estimation equation

$$\mathbf{y} = \mathbf{\Phi}_k \hat{\boldsymbol{\theta}}_k + \boldsymbol{\varepsilon}_k. \tag{44}$$

To estimate the ARMA parameters, $\hat{\boldsymbol{\theta}}_k$, in the above formulation, the least squares operation is performed as follows:

$$\hat{\boldsymbol{\theta}}_k = \left(\mathbf{\Phi}_k^T \mathbf{\Phi}_k\right)^{-1} \mathbf{\Phi}_k^T \mathbf{y}. \tag{45}$$

The performance of a model $\hat{\boldsymbol{\theta}}_k$, for a given k, is evaluated based on its MDL value (which is calculated from its resulting residual error norm). It is of great interest, therefore, to develop a means of efficiently calculating $\boldsymbol{\varepsilon}_k$ when trying to compare various models. If the AR and MA parameters are placed within $\hat{\boldsymbol{\theta}}_k$ in the order that they will be removed from the overspecified model (i.e., the first parameter removed is in the last row of $\hat{\boldsymbol{\theta}}_k$ and the last parameter removed is in the first row of $\hat{\boldsymbol{\theta}}_k$), then we will show that the residual error norm for each candidate model can be calculated recursively!

Rather than solving for the parameters directly with the least squares operation above, we will make use of the 'reduced' QR decomposition of $\mathbf{\Phi}_k$ [27]

$$\mathbf{\Phi}_k = \mathbf{Q}_k \mathbf{R}_k. \tag{46}$$

Briefly, $\mathbf{Q}_k$ is a matrix consisting of k orthonormal columns, i.e.,

$$\mathbf{Q}_k^T \mathbf{Q}_k = \mathbf{I} \tag{47}$$

while $\mathbf{R}_k$ is a square, upper triangular matrix with k rows and columns. Direct substitution of (46) into (45) leads to

$$\hat{\boldsymbol{\theta}}_k = \mathbf{R}_k^{-1} \mathbf{Q}_k^T \mathbf{y}. \tag{48}$$

If we then combine (44), (46), and (48), the following expression is obtained

$$\boldsymbol{\varepsilon}_k = \mathbf{y} - \mathbf{Q}_k \mathbf{Q}_k^T \mathbf{y}. \tag{49}$$

To obtain the recursive relationship desired, we need simply exploit the following fact

$$\mathbf{Q}_k \mathbf{Q}_k^T = \begin{bmatrix} | & & | \\ \mathbf{q}_1 & \cdots & \mathbf{q}_k \\ | & & | \end{bmatrix} \begin{bmatrix} - & \mathbf{q}_1^T & - \\ & \vdots & \\ - & \mathbf{q}_k^T & - \end{bmatrix} = \sum_{i=1}^{k} \mathbf{q}_i \mathbf{q}_i^T \tag{50}$$

where $\mathbf{q}_i$ denotes the i^{th} column of $\mathbf{Q}_k$. The above expression leads to a slight manipulation of (49)

$$\boldsymbol{\varepsilon}_k = \mathbf{y} - \sum_{i=1}^{k} \mathbf{q}_i \mathbf{q}_i^T \mathbf{y} \tag{51}$$

from which follows the recursive relation

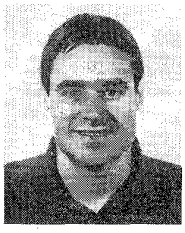
$$\boldsymbol{\varepsilon}_{k-1} = \mathbf{y} - \sum_{i=1}^{k-1} \mathbf{q}_i \mathbf{q}_i^T \mathbf{y} = \boldsymbol{\varepsilon}_k + \mathbf{q}_k \mathbf{q}_k^T \mathbf{y}. \tag{52}$$

ACKNOWLEDGMENT

The authors are grateful for the valuable suggestions given by J. Mathias, T. Mullen, W. C. Karl, M. Appel, K. Yana, and the reviewers.

REFERENCES

[1] L. Ljung, *System Identification: Theory for the User.* Englewood Cliffs, NJ: Prentice Hall, 1987.
[2] W. M. Siebert, *Circuits, Signals, and Systems.* Cambridge, MA: MIT Press, 1986.
[3] S. Cerutti, G. Chiarenza, D. Liberati, P. Mascellani, and G. Pavesi, "A parametric method of identification of single-trial event-related potentials in the brain," *IEEE Trans. Biomed. Eng.*, vol. 35, no. 9, pp. 701–711, Sept. 1988.
[4] J. J. Hollkamp and S. M. Batill, "Automated parameter identification and order reduction for discrete time series models," *AIAA J.*, vol. 29, no. 1, pp. 96–103, Jan. 1991.
[5] A. Papoulis, *Probability, Random Variables, and Stochastic Processes.* New York, NY: McGraw-Hill, 1984.
[6] S. K. Mitra and J. F. Kaiser, *Handbook for digital signal processing.* New York: Wiley, 1993.
[7] G. Strang, *Linear Algebra and its Applications* 3rd ed. San Diego: Harcourt Brace Jovanovich, 1988.
[8] 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.
[9] S. M. Batill and J. J. Hollkamp, "Parameter identification of discrete-time series models for structural response prediction," *AIAA J.*, vol. 27, no. 11, pp. 1636–1643, Nov. 1989.
[10] B. Wahlberg, "Model reductions of high-order estimated models: the asymptotic ML approach," *Int. J. Control*, vol. 49, no. 1, pp. 169–192, 1989.
[11] G. Liang, M. D. Wilkes, and J. A. Cadzow, "ARMA model order estimation based on the eigenvalues of the covariance matrix," *IEEE Trans. Signal Processing*, vol. 41, no. 10, pp. 3003–3009, Oct. 1993.
[12] K. Yana, J. P. Saul, R. D. Berger, M. H. Perrott, and R. J. Cohen, "A time domain approach for the fluctuation analysis of heart rate related to instantaneous lung volume," *IEEE Trans. Biomed. Eng.*, vol. 40, no. 1, pp. 74–81, Jan. 1993.
[13] L. Ljung, *System Identification Toolbox for use with MATLAB.* South Natick, MA: The Math Works, 1988.
[14] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Automat. Contr.*, vol. AC-19, no. 6, pp. 716–722, Dec. 1974.
[15] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, pp. 465–471, 1978.
[16] E. Hannan, "The estimation of the order of an ARMA process," *Ann. Stat.*, vol. 8, pp. 1071–1081, 1980.
[17] T. Soderstrom and P. Stoica, *System Identification.* Englewood Cliffs, NJ: Prentice-Hall, 1989.
[18] J. Rissanen, *Consistent order estimates of autoregressive processes by shortest description of data.* Analysis and Optimization of Stochastic Systems. New York: Academic, 1980. pp. 451-461.
[19] E. Hannan and B. Quinn, "The determination of the order of an autoregression," *J. Roy. Stat. Soc.*, vol. 41, pp. 190–195, 1979.
[20] M. Wax and T. Kailath, "Detection of signals by information theoretic criteria," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 387–392, 1985.
[21] M. H. Perrott, "An efficient ARX model selection procedure applied to autonomic heart rate variability," masters thesis, M. I. T., 1992.
[22] T. Parks and C. Burrus, *Digital Filter Design* New York: Wiley, 1987. p. 245.
[23] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes in C.* Cambridge, MA: MIT Press, 1986.
[24] J. P. Saul, R. D. Berger, P. Albrecht, and R. J. Cohen, "Transfer function analysis of the circulation: unique insights into cardiovascular regulation," *The Amer. Physiol. Soc.*, pp. H1231–H1245, 1991.
[25] M. L. Appel, J. P. Saul, R. D. Berger, and R. J. Cohen, "Closed-loop identification of cardiovascular regulatory mechanisms," *Comput. Cardiol.*, vol. 16, pp. 3–8, 1990.
[26] J. P. Saul, R. D. Berger, M. H. Chen, and R. J. Cohen, "Transfer function analysis of autonomic regulation: Ii — respiratory sinus arrhythmia," *Amer. J. Physiol*, vol. 256, pp. H153–H161, 1989.
[27] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems.* NJ: Englewood Cliffs, 1974.



Michael H. Perrott was born in Austin, TX, in 1967. He received the B.S.E.E. degree from New Mexico State University in 1988, and the M.S.E.E. degree from Massachusetts Institute of Technology in 1992.

He has obtained experience in circuit design and software development at Hughes Aircraft, Sandia National Laboratories, Hewlett-Packard, and Analog Devices. He is currently pursuing the Ph.D. degree in electrical engineering at Massachusetts Institute of Technology. His research involves the application of signal processing and mixed-signal circuit design to wireless communication systems

Richard J. Cohen (M'85) was born in Boston, MA, 1951. He received the B.A. degree in chemistry and physics from Harvard University in 1971, the Ph.D. degree in physics from the Massachusetts Institute of Technology in 1976, and the M.D. degree from Harvard Medical School in 1976. He completed clinical training in internal medicine at the Brigham Women's Hospital in 1979.

He is currently a Professor in the Harvard University-Massachusetts Institute of Technology Division of Health Sciences and Technology and Director of the Harvard-MIT Center for Biomedical Engineering and the NASA Center for Quantitative Physiology, Modeling and Data Analysis. His research interests include cardiovascular physiology and macromolecular interactions.