Parameter Estimation for Autoregressive Gaussian-Mixture Processes: The EMAX Algorithm

Shawn M. Verbout, James M. Ooi, Jeffrey T. Ludwig, and Alan V. Oppenheim, *Fellow, IEEE*

***Abstract*—The problem of estimating parameters of discrete-time non-Gaussian autoregressive (AR) processes is addressed. The subclass of such processes considered is restricted to those whose driving noise samples are statistically independent and identically distributed according to a Gaussian-mixture probability density function (pdf). Because the likelihood function for this problem is typically unbounded in the vicinity of undesirable, degenerate parameter estimates, the maximum likelihood approach is not fruitful. Hence, an alternative approach is taken whereby a finite local maximum of the likelihood surface is sought. This approach, which is termed the quasimaximum likelihood (QML) approach, is used to obtain estimates of the AR parameters as well as the means, variances, and weighting coefficients that define the Gaussian-mixture pdf. A technique for generating solutions to the QML problem is derived using a generalized version of the expectation-maximization principle. This technique, which is referred to as the EMAX algorithm, is applied in four illustrative examples; its performance is compared directly with that of previously proposed algorithms based on the same data model and that of conventional least-squares techniques.**

***Index Terms*— Autoregressive process, iterative methods, parameter estimation.**

I. Introduction

ESTIMATION of parameters of discrete-time non-Gaussian autoregressive (AR) processes has typically been accomplished using methods based on higher order statistics (HOS) (see, for example, [9]–[11] and associated references). These methods are generally robust in the presence of observation noise, are fairly easy to implement, and make few assumptions about the probability density function (pdf) of the AR process. However, because they extract much of their information about the observed process by computing sample moments or cumulants above second order, HOS-based methods tend to produce high-variance estimates when the length of the data record is small. The approach developed in this paper is fundamentally different from the HOS-based approach in that it assumes a specific form for the pdf of the observed data and is therefore entirely parametric. In particular, we consider processes that can be represented as the output of a linear time-invariant (LTI) AR system driven by noise samples that are statistically independent and identically distributed (i.i.d.) according to a Gaussian-mixture pdf (i.e., a pdf that is a weighted average of a finite number of Gaussian densities having arbitrary means and variances); we call such processes autoregressive Gaussian-mixture (ARGMIX) processes.

We seek estimates of the AR parameters jointly with the mixture parameters—i.e., the means, variances, and weighting coefficients—that define the Gaussian-mixture pdf. Joint maximum likelihood (ML) estimates have not been directly pursued in the past because the value of the likelihood function is infinite for certain known, degenerate parameter values. In general, these parameter values are not useful as estimates, even though, strictly speaking, they do maximize the likelihood function. However, as we shall see in this paper, strategies based on finding nondegenerate *local* maxima of the likelihood function yield solutions that are useful. Indeed, Titterington [17] showed that the approach of locally maximizing the likelihood function is useful for the problem of estimating only the mixture parameters, i.e., the problem in which the LTI system is known to be an identity system. He empirically studied the performance of several numerical hill-climbing algorithms for computing ML estimates of the mixture parameters and obtained useful answers. We call the approach based on searching for finite local maxima the quasi-maximum likelihood (QML) approach.

The Gaussian-mixture model is capable of closely approximating many densities and has been considered by a number of researchers for this purpose (see, for example, [3], [4], [8], [14], [17]). Yet apparently only a few researchers, most notably Sengupta and Kay [16] and Zhao [19], have considered Gaussian-mixture models in conjunction with AR systems. Sengupta and Kay [16] address the problem of ML estimation of AR parameters for ARGMIX processes in which two Gaussian pdf's constitute the mixture, each with zero mean and known variance, but with unknown relative weighting. They use a conventional Newton–Raphson optimization algorithm that is initialized by the least-squares solution to find ML estimates for the AR parameters and the single weighting coefficient and show that the performance of the ML estimate is superior to that of the standard forward-backward least-squares method. In a separate investigation, Zhao *et al.* [19] also consider ML estimation of the AR parameters of ARGMIX processes and derive a set of linear equations whose solution gives the ML estimate for the AR parameters when all the mixture parameters are *known*. When

Manuscript received December 2, 1996; revised March 5, 1998. This work was supported in part by the Department of the Navy, Office of the Chief of Naval Research, under Contract N00014-93-1-0686, by the U.S. Air Force Office of Scientific Research under Grant AFOSR-F49620-96-1-0072, and through collaborative participation in the Advanced Sensors Consortium sponsored by the U.S. Army Research Laboratory under Cooperative Agreement DAAL01-96-2-0001. The associate editor coordinating the review of this paper and approving it for publication was Prof. Luis F. Chaparro.

The authors are with the Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139 USA.

Publisher Item Identifier S 1053-587X(98)07075-5.

the mixture parameters are *unknown*, they combine these linear equations with a clever *ad hoc* clustering technique to produce an iterative algorithm for obtaining a joint estimate of both the AR parameters and the mixture parameters. They do not guarantee convergence of this algorithm or optimality of the estimate in any sense but demonstrate empirically that the performance of their algorithm is superior to that of cumulant-based methods in certain cases.

We use the expectation-maximization (EM) method to derive an iterative algorithm, which we refer to as the EMAX algorithm, for jointly estimating the AR parameters and mixture parameters of ARGMIX processes. The EMAX algorithm finds local maxima of the likelihood function. We demonstrate that when initialized appropriately, the estimates corresponding to these local maxima are desirable solutions and, hence, that the likelihood function can still guide us to useful answers via its local maxima even though the ML estimation problem is degenerate.

The paper is organized in the following way: In Section II, we introduce our data model and formulate the QML problem. In Section III, we give a brief overview of the EM and generalized EM algorithms and then use the EM theory to derive the formulas that constitute the EMAX algorithm. This algorithm is proposed as a practical solution to the QML problem. In Section IV, we discuss four distinct applications of the EMAX algorithm and, through computer simulations, compare the performance of the algorithm to that of previously developed algorithms based on a similar data model as well as to that of the standard least-squares technique. Finally, in Section V, we discuss the advantages, limitations, and possible extensions of our method.

II. Problem Formulation

In this section, we present a mathematical model for the random process under consideration, define the set of model parameters to be estimated from a realization of the process, and state criteria that must be met by the most desirable estimates of these model parameters. We begin by introducing some notation that will be used throughout the paper.

A. Notation

We adopt the convention of writing random variables in upper case and particular realizations of random variables in lower case. If X is a random variable, then we denote its pdf by $f_X(\cdot)$. If X takes values from a set containing finitely many elements, its pdf will contain impulses (i.e., Dirac delta functions), but in such cases, this pdf will be used only under appropriate integrals. If Y is also a random variable, then the conditional pdf of X given Y is written $f_{X|Y}(\cdot|\cdot)$. If these densities depend on a parameter θ, then they are written as $f_X(\cdot;\theta)$ and $f_{X|Y}(\cdot|\cdot;\theta)$, respectively. Expectations and conditional expectations associated with densities that depend on a parameter θ are analogously denoted by $E\{\cdot;\theta\}$ and $E\{\cdot|\cdot;\theta\}$, respectively. Vector-valued variables, both random and deterministic, are written in boldface. If $\boldsymbol{x}$ is an n-dimensional vector, then the ith element of $\boldsymbol{x}$ is denoted by $x_{(i)}$ for $i = 1, \cdots, n$. Finally, we introduce the function definition

$$\mathcal{N}(v;\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\ \exp\left\{-\frac{(v-\mu)^2}{2\sigma^2}\right\} \qquad -\infty < v < \infty \tag{1}$$

as a special notation for the Gaussian pdf since this density is used frequently in the remaining sections.

B. Data Model

We consider a discrete-time scalar-valued random process $\{Y_t\}$ that satisfies the Kth-order autoregressive difference equation

$$Y_t = \sum_{k=1}^{K} a_k Y_{t-k} + V_t \tag{2}$$

where $\{a_k\}_{k=1}^{K}$ are the real-valued AR coefficients of the process, and $\{V_t\}$ is a sequence (termed the driving process or driving noise) that consists of i.i.d. random variables having a Gaussian-mixture pdf defined by

$$f_V(v) = \sum_{i=1}^{M} \rho_i\, \mathcal{N}(v;\mu_i,\sigma_i) \tag{3}$$

where the weighting coefficients $\{\rho_i\}_{i=1}^{M}$ satisfy $\rho_i \geq 0$ for $i = 1, 2, \cdots, M$ and $\Sigma_{i=1}^{M}\ \rho_i = 1$. Alternatively, we can express the tth sample of the driving process as

$$V_t = \sigma(\Phi_t)W_t + \mu(\Phi_t) \tag{4}$$

where $\{W_t\}$ is a sequence of i.i.d., zero-mean, unit-variance Gaussian random variables, σ and μ are mappings defined by $\sigma(i) = \sigma_i$ and $\mu(i) = \mu_i$ for $i = 1, 2, \cdots, M$, $\{\Phi_t\}$ is a sequence of i.i.d. discrete-valued random variables distributed according to the probability law $\Pr(\Phi_t = i) = \rho_i$ for $i = 1, 2, \cdots, M$, and the processes $\{W_t\}$ and $\{\Phi_t\}$ are assumed statistically independent. The representation of the driving process given in (4) will be very useful in the derivation of the EMAX algorithm in Section III.

We assume that the order of the autoregression K and the number of constituent densities in the Gaussian-mixture pdf M are given and that the parameters $\boldsymbol{\mu} = [\mu_1 \cdots \mu_M]^T, \boldsymbol{\sigma} = [\sigma_1 \cdots \sigma_M]^T, \boldsymbol{\rho} = [\rho_1 \cdots \rho_M]^T$, and $\boldsymbol{a} = [a_1 \cdots a_K]^T$ are unknown. We observe that the random variables $Y_{-K}, \cdots, Y_{N-1}$ assume the values $y_{-K}, \cdots, y_{N-1}$, respectively, and we wish to estimate the parameter vector

$$\boldsymbol{\Psi} = (\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\rho}, \boldsymbol{a}) \tag{5}$$

based on this observation. For notational convenience, we define the random vectors $\boldsymbol{Y} = [Y_0 \cdots Y_{N-1}]^T$ and $\boldsymbol{Y}_t = [Y_{t-1} \cdots Y_{t-K}]^T$ for $t = 0, 1, \cdots, N$, and denote the realizations of these vectors by $\boldsymbol{y}$ and $\boldsymbol{y}_t$, respectively.

C. Approach to Parameter Estimation

As mentioned earlier, we are not strictly seeking an ML estimate because in most cases, degenerate estimates exist that have infinite likelihood. To see how such degenerate estimates can arise, we can easily verify that if we put, say, $\hat{a}_i = 0$ for $i = 1, \cdots, K, (\hat{\mu}_i, \hat{\sigma}_i, \hat{\rho}_i) = (0, 1, 1/M)$ for $i = 2, \cdots, M$, and $\hat{\mu}_1 = y_0$ and then let $\hat{\sigma}_1 \rightarrow 0$, then the likelihood function $f_{\boldsymbol{Y}_0, \boldsymbol{Y}}(\boldsymbol{y}_0, \boldsymbol{y}; \boldsymbol{\Psi}')$ will increase without bound. This assignment of parameter values corresponds to choosing the unknown AR system to be an identity system and one of the Gaussian densities in the mixture to be an impulse centered directly on one of the observations.

It is apparent from (3) that degenerate estimates are obtained only if one or more of the standard deviation estimates is chosen to be zero. We may be tempted to avoid this problem by restricting all of the standard deviation estimates to be greater than some prespecified positive threshold. However, if this minimum threshold is set too low, then meaningless estimates can arise when the largest likelihood value occurs on the boundary of the restricted parameter space near a singularity at which $\hat{\sigma}_i = 0$ for some i. Yet if the threshold is set too high, we risk excluding the best available estimate since a component of the true Gaussian-mixture pdf may have a standard deviation smaller than the artificially set threshold.

One alternative to maximizing the likelihood function is to find the parameters that achieve the largest of the finite local maxima [3]. In general, no closed-form solution exists for this estimate, and a numerical method must typically be used. Because the likelihood surface may have a very large number of local maxima, we have no guarantee that classical optimization techniques will find the largest local maximum. Yet Titterington [17] has found that methods based on finding local maxima (not necessarily the largest finite local maximum) yield useful estimates. Accordingly, we take the approach of searching for local maxima of the likelihood function using the generalized EM algorithm.

More formally, if we let $\mathcal{P}$ denote the set of all possible values for the parameter vector $\boldsymbol{\Psi}$, then the estimate we seek for $\boldsymbol{\Psi}$ is any $\hat{\boldsymbol{\Psi}}$ satisfying

$$\begin{aligned}\hat{\boldsymbol{\Psi}} &\in \overline{\arg \max_{\boldsymbol{\Psi}' \in \mathcal{P}}} \{\log\, f_{\boldsymbol{Y}_0, \boldsymbol{Y}}(\boldsymbol{y}_0, \boldsymbol{y}; \boldsymbol{\Psi}')\} &(6)\\ &= \overline{\arg \max_{\boldsymbol{\Psi}' \in \mathcal{P}}} \{\log\, f_{\boldsymbol{Y}_0}(\boldsymbol{y}_0; \boldsymbol{\Psi}') + \log\, f_{\boldsymbol{Y}|\boldsymbol{Y}_0}(\boldsymbol{y}|\boldsymbol{y}_0; \boldsymbol{\Psi}')\} &(7)\end{aligned}$$

where the notation $\overline{\arg \max_{x \in \mathcal{P}}} \{g(x)\}$ stands for the set of all parameter values in $\mathcal{P}$ achieving finite local maxima of g.

Since the estimate $\hat{\boldsymbol{\Psi}}$ is defined in terms of the likelihood function but is not obtained through a standard global maximization, we refer to this estimate as a quasimaximum likelihood (QML) estimate. In the sequel, we shall assume that $N \gg K$, i.e., that the number of samples in the observed sequence is much greater than the number of AR parameters to be estimated. Under this assumption, we may, as is standard in the derivation of ML estimates for Gaussian AR processes, ignore the first term of the log-likelihood function appearing on the right-hand side of (7) and assume that a QML estimate is any $\hat{\boldsymbol{\Psi}}$ satisfying

$$\hat{\boldsymbol{\Psi}} \in \overline{\arg \max_{\boldsymbol{\Psi}' \in \mathcal{P}}} \{\log\, f_{\boldsymbol{Y}|\boldsymbol{Y}_0}(\boldsymbol{y}|\boldsymbol{y}_0; \boldsymbol{\Psi}')\}. \tag{8}$$

III. Solution via the EM Principle

In this section, we first review the theory behind the EM and generalized EM (GEM) methods and then use the GEM method to derive the equations that constitute the EMAX algorithm.

A. Theory of the EM and GEM Algorithms

The EM and GEM algorithms, which were first proposed by Dempster *et al.* [1], are iterative techniques for finding local maxima of likelihood functions. Although their convergence rates are slow, these algorithms converge reliably to local maxima of the likelihood function under appropriate conditions, require no derivatives of the likelihood function, and often yield equations that have an intuitively pleasing interpretation.

The EM and GEM algorithms are best suited to problems in which there is a "complete" data specification $\boldsymbol{Z}$ from which the original observations $\boldsymbol{Y}$ can be derived and such that the expectation $E\{\log\, f_{\boldsymbol{Z}}(Z; \boldsymbol{\Psi}')|\boldsymbol{Y} = \boldsymbol{y}; \boldsymbol{\Psi}''\}$ can be easily computed for any two parameter vectors $\boldsymbol{\Psi}', \boldsymbol{\Psi}'' \in \mathcal{P}$. For our problem, we use the complete data specification $\boldsymbol{Z} = (\boldsymbol{Y}, \boldsymbol{\Phi})$, where $\boldsymbol{\Phi}$ is the vector of pdf-selection variables defined by $\boldsymbol{\Phi} = [\Phi_0 \cdots \Phi_{N-1}]^T$. With this choice of complete data, the EM algorithm as applied to our problem generates a sequence of estimates $\{\boldsymbol{\Psi}^{(s)}\}_{s=1}^{\infty}$ according to the recursive formula

$$\begin{aligned}\boldsymbol{\Psi}^{(s+1)} = \arg \max_{\boldsymbol{\Psi}' \in \mathcal{P}} E\{&\log\, f_{\boldsymbol{Y}, \boldsymbol{\Phi}|\boldsymbol{Y}_0}(\boldsymbol{Y}, \boldsymbol{\Phi}|\boldsymbol{y}_0; \boldsymbol{\Psi}')|\\ &\cdot \boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{Y}_0 = \boldsymbol{y}_0; \boldsymbol{\Psi}^{(s)}\}\end{aligned} \tag{9}$$

where some starting estimate $\boldsymbol{\Psi}^{(0)}$ must be chosen to initialize the recursion. We now show that the sequence of estimates $\{\boldsymbol{\Psi}^{(s)}\}_{s=0}^{\infty}$ defined above satisfies the inequality

$$\log\, f_{\boldsymbol{Y}|\boldsymbol{Y}_0}(\boldsymbol{y}|\boldsymbol{y}_0; \boldsymbol{\Psi}^{(s+1)}) \geq \log\, f_{\boldsymbol{Y}|\boldsymbol{Y}_0}(\boldsymbol{y}|\boldsymbol{y}_0; \boldsymbol{\Psi}^{(s)}) \tag{10}$$

for $s = 0, 1, 2, \cdots$; that is, we show that the log-likelihood value associated with our updated parameter estimate is increased at each iteration. We begin by writing the log-likelihood function for the observed data with parameters $\boldsymbol{\Psi}' \in \mathcal{P}$ as

$$\begin{aligned}&\log\, f_{\boldsymbol{Y}|\boldsymbol{Y}_0}(\boldsymbol{y}|\boldsymbol{y}_0; \boldsymbol{\Psi}')\\ &\quad = \log\, f_{\boldsymbol{Y}, \boldsymbol{\Phi}|\boldsymbol{Y}_0}(\boldsymbol{y}, \boldsymbol{\phi}|\boldsymbol{y}_0; \boldsymbol{\Psi}')\\ &\qquad - \log\, f_{\boldsymbol{\Phi}|\boldsymbol{Y}, \boldsymbol{Y}_0}(\boldsymbol{\phi}|\boldsymbol{y}, \boldsymbol{y}_0; \boldsymbol{\Psi}').\end{aligned} \tag{11}$$

Integrating both sides of (11) with respect to $\boldsymbol{\phi}$ against the density $f_{\boldsymbol{\Phi}|\boldsymbol{Y}, \boldsymbol{Y}_0}(\boldsymbol{\phi}|\boldsymbol{y}, \boldsymbol{y}_0; \boldsymbol{\Psi}^{(s)})$ gives

$$\begin{aligned}&\log\, f_{\boldsymbol{Y}|\boldsymbol{Y}_0}(\boldsymbol{y}|\boldsymbol{y}_0; \boldsymbol{\Psi}')\\ &\quad = E\{\log\, f_{\boldsymbol{Y}, \boldsymbol{\Phi}|\boldsymbol{Y}_0}(\boldsymbol{y}, \boldsymbol{\Phi}|\boldsymbol{y}_0; \boldsymbol{\Psi}')|\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{Y}_0 = \boldsymbol{y}_0; \boldsymbol{\Psi}^{(s)}\}\\ &\qquad - E\{\log\, f_{\boldsymbol{\Phi}|\boldsymbol{Y}, \boldsymbol{Y}_0}(\boldsymbol{\Phi}|\boldsymbol{y}, \boldsymbol{y}_0; \boldsymbol{\Psi}')|\boldsymbol{Y} = \boldsymbol{y}\\ &\qquad\quad \boldsymbol{Y}_0 = \boldsymbol{y}_0; \boldsymbol{\Psi}^{(s)}\} &(12)\\ &\quad \triangleq U(\boldsymbol{\Psi}', \boldsymbol{\Psi}^{(s)}) - V(\boldsymbol{\Psi}', \boldsymbol{\Psi}^{(s)}) &(13)\end{aligned}$$

where the functions U and V are defined in the obvious way. Then, (9) can be written as

$$\boldsymbol{\Psi}^{(s+1)} = \arg \max_{\boldsymbol{\Psi}' \in \mathcal{P}} U(\boldsymbol{\Psi}', \boldsymbol{\Psi}^{(s)}). \tag{14}$$

The definition of V together with Jensen's inequality allows us to conclude that $V(\boldsymbol{\Psi}', \boldsymbol{\Psi}^{(s)}) \le V(\boldsymbol{\Psi}^{(s)}, \boldsymbol{\Psi}^{(s)})$ for any $\boldsymbol{\Psi}' \in \mathcal{P}$. Hence, we have

$$\begin{aligned}\log f_{\boldsymbol{Y}|\boldsymbol{Y}_0}(\boldsymbol{y}|\boldsymbol{y}_0; \boldsymbol{\Psi}')|_{\boldsymbol{\Psi}'=\boldsymbol{\Psi}^{(s+1)}} &= U(\boldsymbol{\Psi}^{(s+1)}, \boldsymbol{\Psi}^{(s)}) - V(\boldsymbol{\Psi}^{(s+1)}, \boldsymbol{\Psi}^{(s)}) &(15)\\ &\ge U(\boldsymbol{\Psi}^{(s+1)}, \boldsymbol{\Psi}^{(s)}) - V(\boldsymbol{\Psi}^{(s)}, \boldsymbol{\Psi}^{(s)}) &(16)\\ &\ge U(\boldsymbol{\Psi}^{(s)}, \boldsymbol{\Psi}^{(s)}) - V(\boldsymbol{\Psi}^{(s)}, \boldsymbol{\Psi}^{(s)}) &(17)\\ &= \log f_{\boldsymbol{Y}|\boldsymbol{Y}_0}(\boldsymbol{y}|\boldsymbol{y}_0; \boldsymbol{\Psi}')|_{\boldsymbol{\Psi}'=\boldsymbol{\Psi}^{(s)}} &(18)\end{aligned}$$

which implies that the EM algorithm gives a sequence of parameter estimates with increasing likelihoods. If the function U is continuous in both of its arguments, the sequence of estimates converges to a stationary point of the log-likelihood function [18].

The GEM algorithm is an alternative form of the EM algorithm that is often easier to implement. Such an algorithm chooses $\boldsymbol{\Psi}^{(s+1)}$ such that

$$U(\boldsymbol{\Psi}^{(s+1)}, \boldsymbol{\Psi}^{(s)}) \ge U(\boldsymbol{\Psi}^{(s)}, \boldsymbol{\Psi}^{(s)}) \tag{19}$$

at each iteration s. It does not necessarily select $\boldsymbol{\Psi}^{(s+1)}$ such that (14) is satisfied. Using the same reasoning we used to go from (15)–(18), we see that a GEM algorithm also produces a sequence of parameter estimates with increasing likelihoods. Whether the limit of this sequence of estimates is a stationary point of the likelihood function depends on the particular rule for selecting $\boldsymbol{\Psi}^{(s+1)}$ from $\boldsymbol{\Psi}^{(s)}$. If $\boldsymbol{\Psi}^{(s+1)}$ is selected so that it is a *local* maximum of $U(\boldsymbol{\Psi}', \boldsymbol{\Psi}^{(s)})$ over $\boldsymbol{\Psi}' \in \mathcal{P}$, then the sequence converges to a stationary point of the likelihood function [6], [18]. We will use this local-maximum rule for selecting updated parameters in our GEM algorithm. As is the case with all "hill-climbing" algorithms, the limit of the sequence of estimates generated by an EM or GEM algorithm may not be a global maximum of the likelihood function. Therefore, choosing $\boldsymbol{\Psi}^{(0)}$ judiciously is the key to obtaining a good parameter estimate. A simple method for choosing $\boldsymbol{\Psi}^{(0)}$ is given and empirically shown to be adequate in Section IV.

B. The EMAX Algorithm

In this section, we give for our problem explicit equations that define the EMAX algorithm. The EMAX algorithm is derived by using a GEM method that chooses $\boldsymbol{\Psi}^{(s+1)}$ to be a local maximum of $U(\boldsymbol{\Psi}', \boldsymbol{\Psi}^{(s)})$ over $\boldsymbol{\Psi}' \in \mathcal{P}$.

To derive the EMAX algorithm, we let $\boldsymbol{\Psi}' = (\boldsymbol{\mu}', \boldsymbol{\sigma}', \boldsymbol{\rho}', \boldsymbol{a}')$ and write (14) as

$$\begin{aligned}\boldsymbol{\Psi}^{(s+1)} = \arg \max_{\boldsymbol{a}', \boldsymbol{\mu}', \boldsymbol{\sigma}', \boldsymbol{\rho}'} & E\{\log f_{\boldsymbol{\Phi}|\boldsymbol{Y}_0}(\boldsymbol{\Phi}|\boldsymbol{y}_0; \boldsymbol{\rho}')\\ &+ \log f_{\boldsymbol{Y}|\boldsymbol{\Phi}, \boldsymbol{Y}_0}\\ &\cdot (\boldsymbol{y}|\boldsymbol{\Phi}, \boldsymbol{y}_0; \boldsymbol{a}', \boldsymbol{\mu}', \boldsymbol{\sigma}')|\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{Y}_0 = \boldsymbol{y}_0; \boldsymbol{\Psi}^{(s)}\}. \end{aligned}\tag{20}$$

This is equivalent to solving the two maximization problems

$$\begin{aligned}\boldsymbol{\rho}^{(s+1)} = \arg \max_{\boldsymbol{\rho}'} & E\\ &\cdot \{\log f_{\boldsymbol{\Phi}|\boldsymbol{Y}_0}(\boldsymbol{\Phi}|\boldsymbol{y}_0; \boldsymbol{\rho}')|\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{Y}_0 = \boldsymbol{y}_0; \boldsymbol{\Psi}^{(s)}\}\end{aligned}\tag{21}$$

and

$$\begin{aligned}(\boldsymbol{a}^{(s+1)}, \boldsymbol{\mu}^{(s+1)}, \boldsymbol{\sigma}^{(s+1)}) &= \arg \max_{\boldsymbol{a}', \boldsymbol{\mu}', \boldsymbol{\sigma}'} E\\ &\quad \cdot \{\log f_{\boldsymbol{Y}|\boldsymbol{\Phi}, \boldsymbol{Y}_0}(\boldsymbol{y}|\boldsymbol{\Phi}, \boldsymbol{y}_0; \boldsymbol{a}', \boldsymbol{\mu}', \boldsymbol{\sigma}')|\\ &\quad \cdot \boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{Y}_0 = \boldsymbol{y}_0; \boldsymbol{\Psi}^{(s)}\}.\end{aligned}\tag{22}$$

To find $\boldsymbol{\rho}^{(s+1)}$ so that (21) is satisfied, we first define the functions $\{d_j\}_{j=1}^{M}$ and $\{C_j\}_{j=1}^{M}$ by

$$d_j(\phi) = \begin{cases}1 & \text{if } \phi = j,\\ 0 & \text{otherwise}\end{cases} \tag{23}$$

$$C_j([\phi_0 \cdots \phi_{N-1}]^T) = \sum_{t=0}^{N-1} d_j(\phi_t) \tag{24}$$

that is, $C_j(\boldsymbol{\Phi})$ is the number of times the symbol j appears in the vector $\boldsymbol{\Phi}$. In addition, for notational convenience, we define the function $P_{t,j}$ by

$$P_{t,j}(\boldsymbol{\Psi}') = \Pr\{\Phi_t = j|\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{Y}_0 = \boldsymbol{y}_0; \boldsymbol{\Psi}'\} \tag{25}$$

for all $\boldsymbol{\Psi}' \in \mathcal{P}$, for $t = 0, \cdots, N-1$ and $j = 1, \cdots, M$. Using these definitions, the maximization in (21), which is over all $\boldsymbol{\rho}'$ such that $\rho'_{(j)} \ge 0$ and $\Sigma_{j=1}^{M} \rho'_{(j)} = 1$, can be written

$$\begin{aligned}\rho_{(j)}^{(s+1)} &= \arg \max_{\boldsymbol{\rho}'} E\\ &\quad \cdot \left\{\log \prod_{j=1}^{M} \rho'^{C_j(\boldsymbol{\Phi})}_{(j)} |\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{Y}_0 = \boldsymbol{y}_0; \boldsymbol{\Psi}^{(s)}\right\} &(26)\\ &= \arg \max_{\boldsymbol{\rho}'} E\\ &\quad \cdot \left\{\sum_{j=1}^{M} C_j(\boldsymbol{\Phi}) \log \rho'_{(j)} |\boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{Y}_0 = \boldsymbol{y}_0; \boldsymbol{\Psi}^{(s)}\right\} &(27)\\ &= \arg \max_{\boldsymbol{\rho}'} \sum_{j=1}^{M} \sum_{t=0}^{N-1} P_{t,j}(\boldsymbol{\Psi}^{(s)}) \log \rho'_{(j)} &(28)\\ &= \frac{1}{N} \sum_{t=0}^{N-1} P_{t,j}(\boldsymbol{\Psi}^{(s)}) &(29)\end{aligned}$$

where the last equality follows from Jensen's inequality.

To attempt the maximization in (22), we use the knowledge that the driving process is a sequence of i.i.d. Gaussian-mixture random variables to write the pdf for $\boldsymbol{Y}$ conditioned on $\boldsymbol{\Phi}$ and $\boldsymbol{Y}_0$ as

$$\begin{aligned} f_{\boldsymbol{Y}|\boldsymbol{\Phi},\boldsymbol{Y}_0}&(\boldsymbol{y}|\boldsymbol{\Phi},\boldsymbol{y}_0;\boldsymbol{a}',\boldsymbol{\mu}',\boldsymbol{\sigma}') \\ &= \prod_{t=0}^{N-1} \mathcal{N}(y_t-\boldsymbol{y}_t^T\boldsymbol{a}',\mu'_{(\Phi_t)},\sigma'_{(\Phi_t)}) \qquad (30) \\ &= \prod_{t=0}^{N-1} \frac{1}{\sqrt{2\pi}\sigma'_{(\Phi_t)}} \\ &\quad \cdot \exp\left\{-\frac{1}{2\sigma'^2_{(\Phi_t)}}(y_t-\boldsymbol{y}_t^T\boldsymbol{a}'-\mu'_{(\Phi_t)})^2\right\}. \qquad (31) \end{aligned}$$

Notice that the term $y_t - \boldsymbol{y}_t^T\boldsymbol{a}'$ represents the residual or prediction error obtained by using $\boldsymbol{a}'$ as the AR parameter vector. The function being maximized in (22) can then be written as

$$\begin{aligned} E\{\log\, &f_{\boldsymbol{Y}|\boldsymbol{\Phi},\boldsymbol{Y}_0}(\boldsymbol{y}|\boldsymbol{\Phi},\boldsymbol{y}_0;\boldsymbol{a}',\boldsymbol{\mu}',\boldsymbol{\sigma}')|\boldsymbol{Y}=\boldsymbol{y},\boldsymbol{Y}_0=\boldsymbol{y}_0;\boldsymbol{\Psi}^{(s)}\} \\ &= -\frac{N}{2}\log 2\pi - \sum_{t=0}^{N-1}\sum_{j=1}^{M} P_{t,j}(\boldsymbol{\Psi}^{(s)})\log\sigma'_{(j)} \\ &\quad - \sum_{t=0}^{N-1}\sum_{j=1}^{M} P_{t,j}(\boldsymbol{\Psi}^{(s)})\,\frac{(y_t-\boldsymbol{y}_t^T\boldsymbol{a}'-\mu'_{(j)})^2}{2\sigma'^2_{(j)}}. \qquad (32) \end{aligned}$$

Taking derivatives of this expression with respect to the quantities $\boldsymbol{\mu}'$, $\boldsymbol{\sigma}'$, and $\boldsymbol{a}'$ and setting the resulting expressions equal to zero yields three coupled nonlinear equations that define a stationary point of the right-hand side of (32). Because we are unable to solve these nonlinear equations analytically, it is difficult to find a global maximum. We instead use the method of coordinate ascent to numerically find a *local* maximum, resulting in a GEM algorithm rather than an EM algorithm. Coordinate ascent increases a multivariate function at each iteration by changing one variable at a time. If, at each iteration, the variable that is allowed to change is chosen to achieve the maximum of the function while the other variables are kept fixed, then coordinate ascent converges to a local maximum of the function [6]. Coordinate ascent is attractive because it is simple to maximize (32) separately over each variable as

$$\begin{aligned} \arg\max_{\mu'_{(j)}}\, &E\{\log\, f_{\boldsymbol{Y}|\boldsymbol{\Phi},\boldsymbol{Y}_0}(\boldsymbol{y}|\boldsymbol{\Phi},\boldsymbol{y}_0; \\ &(\boldsymbol{a}',\mu'_{(1)},\cdots,\mu'_{(M)},\boldsymbol{\sigma}'))|\boldsymbol{Y}=\boldsymbol{y},\boldsymbol{Y}_0=\boldsymbol{y}_0;\boldsymbol{\Psi}^{(s)}\} \\ &= \frac{\displaystyle\sum_{t=0}^{N-1} P_{t,j}(\boldsymbol{\Psi}^{(s)})(y_t-\boldsymbol{y}_t^T\boldsymbol{a}')}{\displaystyle\sum_{t=0}^{N-1} P_{t,j}(\boldsymbol{\Psi}^{(s)})} \qquad (33) \end{aligned}$$

$$\begin{aligned} \arg\max_{\sigma'_{(j)}}\, &E\{\log\, f_{\boldsymbol{Y}|\boldsymbol{\Phi},\boldsymbol{Y}_0}(\boldsymbol{y}|\boldsymbol{\Phi},\boldsymbol{y}_0; \\ &(\boldsymbol{a}',\boldsymbol{\mu}',\sigma'_{(1)},\cdots,\sigma'_{(M)}))|\boldsymbol{Y}=\boldsymbol{y},\boldsymbol{Y}_0=\boldsymbol{y}_0;\boldsymbol{\Psi}^{(s)}\} \\ &= \sqrt{\frac{\displaystyle\sum_{t=0}^{N-1} P_{t,j}(\boldsymbol{\Psi}^{(s)})(y_t-\boldsymbol{y}_t^T\boldsymbol{a}'-\mu'_{(j)})^2}{\displaystyle\sum_{t=0}^{N-1} P_{t,j}(\boldsymbol{\Psi}^{(s)})}} \qquad (34) \end{aligned}$$

$$\begin{aligned} \arg\max_{\boldsymbol{a}'}\, &E\{\log\, f_{\boldsymbol{Y}|\boldsymbol{\Phi},\boldsymbol{Y}_0}(\boldsymbol{y}|\boldsymbol{\Phi},\boldsymbol{y}_0; \\ &(\boldsymbol{a}',\boldsymbol{\mu}',\boldsymbol{\sigma}'))|\boldsymbol{Y}=\boldsymbol{y},\boldsymbol{Y}_0=\boldsymbol{y}_0;\boldsymbol{\Psi}^{(s)}\} \\ &= \left[\sum_{t=0}^{N-1}\sum_{j=1}^{M}\frac{P_{t,j}(\boldsymbol{\Psi}^{(s)})}{(\sigma'_{(j)})^2}\,\boldsymbol{y}_t\boldsymbol{y}_t^T\right]^{-1} \\ &\quad \cdot \left[\sum_{t=0}^{N-1}\sum_{j=1}^{M}\frac{P_{t,j}(\boldsymbol{\Psi}^{(s)})}{(\sigma'_{(j)})^2}\,(y_t-\mu'_{(j)})\boldsymbol{y}_t\right]. \qquad (35) \end{aligned}$$

Using the equations above, the coordinate-ascent algorithm is described as follows:

INITIALIZATION:

$$\tilde{\mu}^{(0)}_{(j)} = \mu^{(s)}_{(j)}, \qquad j=1,\cdots,M \qquad (36)$$

$$\tilde{\sigma}^{(0)}_{(j)} = \sigma^{(s)}_{(j)}, \qquad j=1,\cdots,M \qquad (37)$$

$$\tilde{\boldsymbol{a}}^{(0)} = \boldsymbol{a}^{(s)}. \qquad (38)$$

ITERATION:

$$\tilde{\mu}^{(i+1)}_{(j)} = \frac{\displaystyle\sum_{t=0}^{N-1} P_{t,j}(\boldsymbol{\Psi}^{(s)})(y_t-\boldsymbol{y}_t^T\tilde{\boldsymbol{a}}^{(i)})}{\displaystyle\sum_{t=0}^{N-1} P_{t,j}(\boldsymbol{\Psi}^{(s)})}, \qquad j=1,\cdots,M \qquad (39)$$

$$\tilde{\sigma}^{(i+1)}_{(j)} = \sqrt{\frac{\displaystyle\sum_{t=0}^{N-1} P_{t,j}(\boldsymbol{\Psi}^{(s)})(y_t-\boldsymbol{y}_t^T\tilde{\boldsymbol{a}}^{(i)}-\tilde{\mu}^{(i+1)}_{(j)})^2}{\displaystyle\sum_{t=0}^{N-1} P_{t,j}(\boldsymbol{\Psi}^{(s)})}} \qquad j=1,\cdots,M \qquad (40)$$

$$\begin{aligned} \tilde{\boldsymbol{a}}^{(i+1)} &= \left[\sum_{t=0}^{N-1}\sum_{j=1}^{M}\frac{P_{t,j}(\boldsymbol{\Psi}^{(s)})}{(\tilde{\sigma}^{(i+1)}_{(j)})^2}\,\boldsymbol{y}_t\boldsymbol{y}_t^T\right]^{-1} \\ &\quad \cdot \left[\sum_{t=0}^{N-1}\sum_{j=1}^{M}\frac{P_{t,j}(\boldsymbol{\Psi}^{(s)})}{(\tilde{\sigma}^{(i+1)}_{(j)})^2}\,(y_t-\tilde{\mu}^{(i+1)}_{(j)})\boldsymbol{y}_t\right]. \qquad (41) \end{aligned}$$

If this recursion is iterated for $i = 0,\cdots,J-1$, then we define our parameter updates by $\boldsymbol{a}^{(s+1)} = \tilde{\boldsymbol{a}}^{(J)}$, $\boldsymbol{\mu}^{(s+1)} = \tilde{\boldsymbol{\mu}}^{(J)}, \boldsymbol{\sigma}^{(s+1)} = \tilde{\boldsymbol{\sigma}}^{(J)}$. For sufficiently large values of J, the updated parameters are, for practical purposes, local maxima of (32). Since the EMAX algorithm is a GEM algorithm that chooses the updated parameter estimates to be local maxima of (32), it converges to a stationary point. In summary, then, a single iteration of the EMAX algorithm consists of computing $\{P_{t,j}(\boldsymbol{\Psi}^{(s)})\}$, applying (29), and iterating (39)–(41) until convergence.

As shown in Fig. 1, the EMAX algorithm can be conceptually decomposed into three main steps, which are iterated to produce the final parameter estimates. Observe that the filter $1 - \Sigma_{i=1}^{K}\, a^{(s)}_{(i)}z^{-i}$ can be interpreted as the current estimate of the inverse of the AR filter. In the first block of Fig. 1, this inverse filter is applied to the observations to produce the residual sequence $v^{(s)}_t = y_t - \boldsymbol{y}_t^T\boldsymbol{a}^{(s)}$, which can be interpreted as an estimate of the driving noise. This

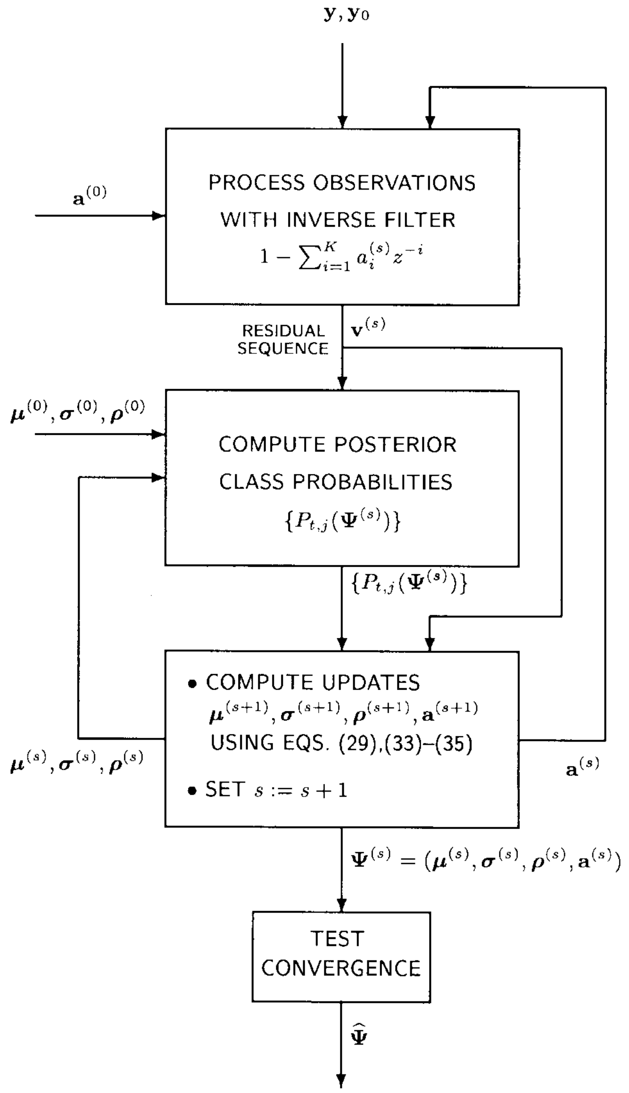

$\mathbf{y}, \mathbf{y}_0$
$\mathbf{a}^{(0)}$
PROCESS OBSERVATIONS WITH INVERSE FILTER $1 - \sum_{i=1}^{K} a_i^{(s)} z^{-i}$
RESIDUAL SEQUENCE
$\mathbf{v}^{(s)}$
$\boldsymbol{\mu}^{(0)}, \boldsymbol{\sigma}^{(0)}, \boldsymbol{\rho}^{(0)}$
COMPUTE POSTERIOR CLASS PROBABILITIES $\{P_{t,j}(\Psi^{(s)})\}$
$\{P_{t,j}(\Psi^{(s)})\}$
• COMPUTE UPDATES $\boldsymbol{\mu}^{(s+1)}, \boldsymbol{\sigma}^{(s+1)}, \boldsymbol{\rho}^{(s+1)}, \mathbf{a}^{(s+1)}$ USING EQS. (29),(33)–(35)
• SET $s := s+1$
$\boldsymbol{\mu}^{(s)}, \boldsymbol{\sigma}^{(s)}, \boldsymbol{\rho}^{(s)}$
$\mathbf{a}^{(s)}$
$\Psi^{(s)} = (\boldsymbol{\mu}^{(s)}, \boldsymbol{\sigma}^{(s)}, \boldsymbol{\rho}^{(s)}, \mathbf{a}^{(s)})$
TEST CONVERGENCE
$\widehat{\Psi}$

Fig. 1. Block diagram representation of the EMAX algorithm developed in Section III.

residual sequence is used to compute the posterior probabilities $\{P_{t,j}(\boldsymbol{\Psi}^{(s)})\}$. Under the hypothesis that $\boldsymbol{a}^{(s)}$ is the true AR parameter vector, these residuals are statistically independent. Using the representation for the driving process given in (4), we may take the view that each sample of the residual sequence is a particular realization arising from one of M randomly chosen classes, where the pdf characterizing the jth of these classes is $\mathcal{N}(\cdot; \mu_{(j)}^{(s)}, \sigma_{(j)}^{(s)})$. For the tth sample of the driving noise sequence, the value of the class label j is determined by the pdf-selection variable Φ_t. Assuming that the mixture parameters are $\boldsymbol{\mu}^{(s)}, \boldsymbol{\sigma}^{(s)}$, and $\boldsymbol{\rho}^{(s)}$, we can easily compute the posterior probability $P_{t,j}(\boldsymbol{\Psi}^{(s)})$ that the tth sample is a realization from class j using Bayes' rule; this is the operation being performed in the second block of Fig. 1. With these posterior probabilities, we first compute the updated estimate of the weighting coefficient vector $\boldsymbol{\rho}^{(s+1)}$ according to (29). We then compute $\boldsymbol{\mu}^{(s+1)}, \boldsymbol{\sigma}^{(s+1)}$ and $\boldsymbol{a}^{(s+1)}$ by iterating (39)–(41) until convergence to some prespecified numerical tolerance is obtained; this operation is represented by the third block. As shown in Fig. 1, the process is repeated, starting again from the first block, until convergence.

A single iteration of (39)–(41) has the following intuitive interpretation. The new estimate for the mean of the jth class is a weighted time average of the residuals, where the weight on the tth residual sample is proportional to the posterior probability that the sample belongs to class j. The new estimate for the variance of the jth class is a weighted time average of the square of residuals with the previously computed estimate of the mean of the jth class removed; once again, the weight on the tth residual sample is proportional to the posterior probability that the sample belongs to class j. The new estimate for the AR coefficient vector is updated via a generalized version of the Yule–Walker equations [19] using the most recent estimates of the means and variances.

We make the final observation that if the values of the parameters in any subset of the $3M$ mixture parameters are known, then the update equations for the parameter estimates can easily be modified, and the properties of the EMAX algorithm will be preserved. Specifically, we simply replace the parameter updates in (29) and (39)–(41) with the corresponding known parameter values. Clearly, updates for the known parameters would not be performed in this case.

IV. Numerical Examples

In this section, we present several examples to illustrate the behavior and performance of the EMAX algorithm. These examples were selected with several objectives in mind:

1) to verify that the EMAX algorithm behaves as expected and produces results consistent with those obtained by others on relevant ML estimation problems;
2) to illustrate that the EMAX algorithm performs significantly better in certain estimation problems than either conventional least-squares techniques or previously proposed algorithms based on a similar data model;
3) to demonstrate that the EMAX algorithm can be used to obtain good approximations to ML estimates in cases where the functional form for the pdf of the driving process is unknown;
4) to show that the EMAX algorithm can be very useful in common signal processing problems where the primary objective is to recover a signal from corrupted measurements.

For each of the examples presented here, we found that the following simple method for generating an initial parameter estimate $\boldsymbol{\Psi}^{(0)} = (\boldsymbol{\mu}^{(0)}, \boldsymbol{\sigma}^{(0)}, \boldsymbol{\rho}^{(0)}, \boldsymbol{a}^{(0)})$ for the EMAX algorithm yielded good average performance. The vector $\boldsymbol{a}^{(0)}$ was computed using the forward-backward least-squares technique from traditional AR signal analysis [7]. Each of the M elements of the mean vector $\boldsymbol{\mu}^{(0)}$ was randomly generated according to a uniform pdf having region of support $[\min_t\{v_{(t)}^{(0)}\}, \max_t\{v_{(t)}^{(0)}\}]$, where $v_{(t)}^{(0)}$ is the tth element of the residual sequence $\boldsymbol{v}^{(0)}$ produced by applying the filter $1 - \Sigma_{i=1}^{K} a_{(i)}^{(0)} z^{-i}$ to the sequence of observations. Each element of $\boldsymbol{\sigma}$ was randomly chosen according to a uniform pdf with region of support $[0, \max_t\{v_{(t)}^{(0)}\} - \min_t\{v_{(t)}^{(0)}\}]$. Finally,

TABLE I

Sample Means and Variances for Parameter Estimates from Example 1. Entries Were Computed Using Results of 5000 Trials for i) the Algorithm of Sengupta and Kay (S-K), ii) the EMAX Algorithm with Known Standard Deviations (EMAX-KSD), and iii) the EMAX Algorithm with Unknown Standard Deviations (EMAX-USD). Cramér–Rao Bounds on the Estimation Variances, as Reported by Sengupta and Kay, Are also Listed for the Case of Known Standard Deviations (KSD's)

	True Value	*Sample Mean* (S-K)	*Sample Mean* (EMAX-KSD)	*Sample Mean* (EMAX-USD)	*Sample Variance* (S-K)	*Sample Variance* (EMAX-KSD)	*Sample Variance* (EMAX-USD)	*Cramèr-Rao Bound* (USD)
a_1	1.352	1.3527	1.3518	1.3518	1.0219×10^{-4}	1.0727×10^{-4}	1.0782×10^{-4}	1.0491×10^{-4}
a_2	-1.338	-1.3391	-1.3378	-1.3377	2.4619×10^{-4}	2.5955×10^{-4}	2.6073×10^{-4}	2.5961×10^{-4}
a_3	0.662	0.6629	0.6619	0.6619	2.4253×10^{-4}	2.6125×10^{-4}	2.6225×10^{-4}	2.5961×10^{-4}
a_4	-0.240	-0.2404	-0.2402	-0.2402	1.0352×10^{-4}	1.0742×10^{-4}	1.0753×10^{-4}	1.0491×10^{-4}
σ^2	10.900	10.8544	10.8963	10.8946	1.2061	1.1655	2.8941	0.3149

the elements of the weighting coefficient vector $\boldsymbol{\rho}^{(0)}$ were all set equal to $1/M$. For special cases in which certain elements of $\boldsymbol{\Psi}$ were assumed known, no initial estimate needed to be chosen.

A. Example 1—Comparison with Previous Work (Part I)

We begin with a simple example for which numerical results have already been reported by Sengupta and Kay [16]. For direct comparison of the performance of our EMAX algorithm to that of the Sengupta–Kay (S-K) algorithm, we have replicated the computer simulations carried out in their previous work. The problem considered by those authors was the ML estimation of the parameters of a fourth-order AR process, whose AR coefficients are given by

$$(a_1, a_2, a_3, a_4) = (1.352, -1.338, 0.662, -0.240). \tag{42}$$

The driving noise for this process was assumed to consist of i.i.d. samples distributed according to the two-component Gaussian-mixture pdf

$$\begin{aligned} f_V(v) = &\rho_1 \, \mathcal{N}(v; \mu_1, \sigma_1) + \rho_2 \, \mathcal{N}(v; \mu_2, \sigma_2) \\ &-\infty < v < \infty \end{aligned} \tag{43}$$

where the mixture parameters $\rho_1, \mu_1, \sigma_1, \rho_2, \mu_2$, and σ_2 are defined by

$$(\rho_1, \mu_1, \sigma_1) = (0.9, 0.0, 1.0) \tag{44}$$

$$(\rho_2, \mu_2, \sigma_2) = (0.1, 0.0, 10.0). \tag{45}$$

A plot of the power spectral density of this process is shown in Fig. 2.

Sengupta and Kay assumed that the values of μ_1, μ_2, σ_1, and σ_2 were known and that the values of the remaining model parameters a_1, a_2, a_3, a_4, and ρ_1 (and, of course, ρ_2 since $\rho_2 = 1 - \rho_1$) were unknown. They developed a Newton–Raphson algorithm for obtaining ML estimates of the AR parameters and of the overall variance σ^2 associated with the driving process, which is given by

$$\sigma^2 = \rho_1 \sigma_1^2 + (1 - \rho_1)\sigma_2^2. \tag{46}$$

Obtaining an ML estimate of σ^2 is, in this case, equivalent to obtaining an unconstrained ML estimate of ρ_1. This is true because the parameters σ^2 and ρ_1 stand in one-to-one correspondence, and the ML estimation procedure is invariant with respect to such invertible transformations on the parameters of the log-likelihood function [12].

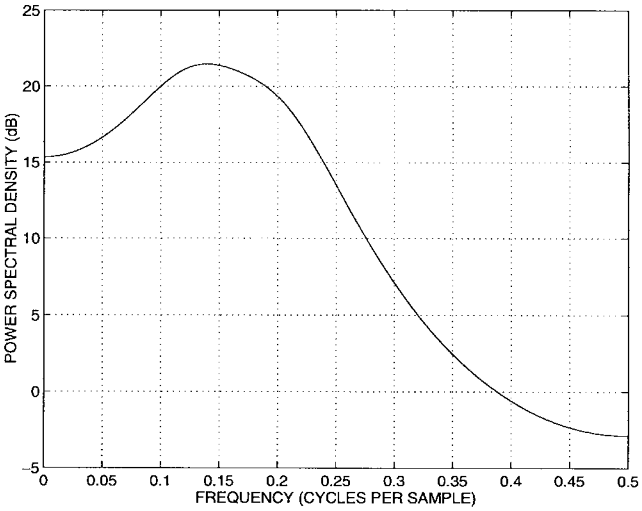


Fig. 2. Power spectral density of fourth-order AR process discussed in Example 1.

As was done in [16], we performed a total of 5000 trials. On each trial, a sequence of 1000 data points was generated and processed using the EMAX algorithm. The sample means and variances of the parameter estimates produced by the EMAX algorithm are presented in Table I in the column labeled EMAX-KSD (where KSD stands for *known standard deviations*). The results of a separate simulation in which the standard deviations were assumed to be unknown are also listed in Table I in the column labeled EMAX USD (where USD stands for *unknown standard deviations*). Remarkably, the sample variance of the estimates of the AR coefficients increased negligibly for the case in which the standard deviations were unknown. However, the sample variance of the estimate of the remaining parameter σ^2 increased dramatically over that for the case in which the standard deviations were known.

We observe from Table I that the estimate of σ^2 produced by the EMAX-KSD algorithm has less bias and a smaller sample variance than the corresponding estimate produced by the S-K algorithm. A possible explanation for this discrepancy is that Sengupta and Kay did not constrain their estimate of ρ_1 (which is a function of σ^2), whereas the EMAX algorithm appropriately constrains its estimate of ρ_1 to be between 0 and 1. We make two further observations from Table I: i) All of the sample means associated with the AR parameter estimates generated by the S-K algorithm exhibit slightly more bias than the sample means generated by the EMAX algorithm; ii) all of the sample variances of these same estimates generated by the S-K algorithm are below the Cramér–Rao bound, whereas only one of the sample variances generated by the EMAX algorithm has this property. These discrepancies may stem from the methodology used by Sengupta and Kay. They report that in approximately 1% of the trials performed for this experiment (i.e., in approximately 50 out of 5000 trials), their Newton–Raphson optimization algorithm did not converge. Whenever convergence was not obtained, the results of the corresponding trial were discarded; hence, these trials are not reflected in the statistics presented in Table I. In contrast, the EMAX algorithm converged in all of the 5000 trials; hence, the results of all trials are represented in the table. The reduction in variance realized by the S-K algorithm over the EMAX algorithm may be due to the discarded trials. This conjecture is plausible if, on those occasions when the Newton–Raphson algorithm did not converge, the ML parameter estimates were relatively far from the true parameter values. If such a correlation exists between events, then it is precisely the estimates that are never obtained because of lack of convergence that distort the sample variances reported by Sengupta and Kay.

B. Example 2—Comparison with Previous Work (Part II)

Our next example illustrates that the EMAX algorithm performs significantly better in certain kinds of estimation problems than the algorithm previously proposed by Zhao *et al.* [19], which is based on precisely the same statistical model for the observed data as that presented in Section II-B. The algorithm of Zhao, which is apparently not motivated in any respect by the EM principle, is similiar in structure to the EMAX algorithm. In particular, both of these iterative algorithms use the same set of generalized normal equations to solve for the estimates of the AR parameters when given the values of the mixture parameters. In addition, at the beginning of each iteration, both algorithms use the resulting AR parameter estimates to inverse filter the observation sequence. The main difference lies in the stage of each algorithm that estimates the pdf mixture parameters from the sequence of residuals. As discussed in Section III, the EMAX algorithm uses the information available in the residual sequence to climb the likelihood surface. In contrast, Zhao abandons a likelihood-based approach (citing a desire to avoid the degenerate solutions mentioned earlier) in favor of a heuristic clustering algorithm.

In the two-component mixture case, the clustering algorithm first sorts the residual samples in ascending order and then seeks out the best point at which to divide these sorted samples into two disjoint sets. The optimum point is defined as that which minimizes the average value of the sample variances associated with these two sets. Once this optimum point is found, Zhao's estimates of the means and variances of the constituent Gaussians are the sample means and sample variances associated with the two sets, and the estimate of the unknown weighting coefficient is simply the fraction of samples contained in the first set with respect to the total number of residual samples.

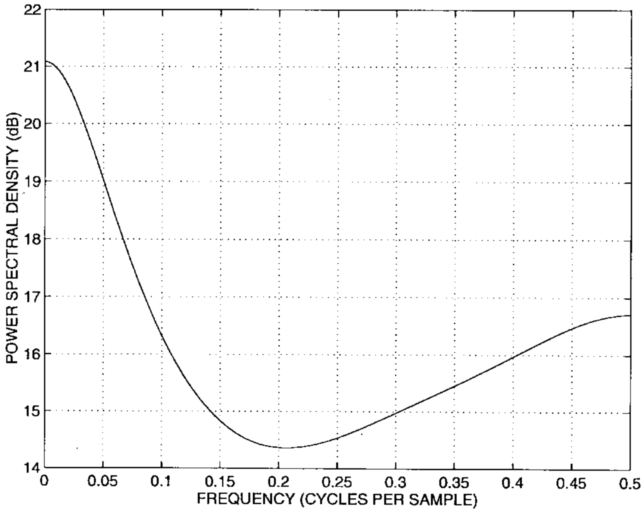


Fig. 3. Power spectral density of fourth-order AR process discussed in Example 2.

We have observed that the algorithm of Zhao does not perform well when the constituent Gaussian densities in the driving-noise pdf have equal means. In this example, we demonstrate that in such a case, the performance of the EMAX algorithm is markedly superior to that of the Zhao algorithm. In particular, we considered the problem of estimating the parameters of an ARGMIX process whose AR coefficients are given by

$$(a_1, a_2, a_3, a_4) = (-0.1000, -0.2238, -0.0844, -0.0294). \tag{47}$$

The pdf for the driving noise in this case was assumed to be a two-component Gaussian-mixture pdf as in (43) but now with mixture parameters defined by

$$(\rho_1, \mu_1, \sigma_1) = (0.6, 0.0, 1.0) \tag{48}$$

$$(\rho_2, \mu_2, \sigma_2) = (0.4, 0.0, 10.0). \tag{49}$$

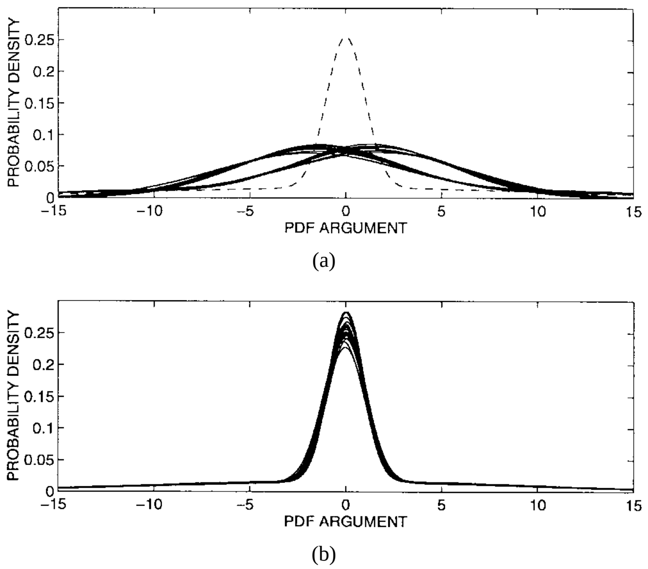
A plot of the power spectral density of this process is shown in Fig. 3.

To compare the performance of the two algorithms, we performed a total of 500 trials. During each trial, a sequence of 1000 data points was generated and processed with the EMAX algorithm and Zhao algorithm. The sample means, variances, and mean square errors of the parameter estimates produced by the two algorithms are presented in Table II. We note that the Zhao algorithm produces strongly biased estimates in this example. In addition, we note that the mean square errors associated with the EMAX algorithm are approximately 25 times smaller than those associated with the Zhao algorithm. Clearly, contributions to the mean square error for Zhao's estimates come not only from the bias term but also from the high variance associated with her estimator.

The difficulties with the Zhao algorithm in this case may be explained by its inability to obtain good mixture parameter estimates. The quality of the mixture parameter estimates is inherently limited because the clustering algorithm essentially assigns the individual densities in the Gaussian mixture to be representatives of disjoint portions of the histogram of the

TABLE II
SAMPLE MEANS, VARIANCES, AND MEAN SQUARE ERROR (MSE) VALUES FOR PARAMETER ESTIMATES OF EXAMPLE 2. ENTRIES WERE COMPUTED USING RESULTS OF 500 TRIALS FOR i) THE ZHAO ALGORITHM AND ii) THE EMAX ALGORITHM. RATIOS OF SAMPLE MSE VALUES (MSE OF ZHAO TO MSE OF EMAX) ARE ALSO GIVEN

	True Value	*Sample Mean* (EMAX)	*Sample Mean* (ZHAO)	*Sample Variance* (EMAX)	*Sample Variance* (ZHAO)	*Sample MSE* (EMAX)	*Sample MSE* (ZHAO)	*Ratio of MSE's*
a_1	-0.1000	-0.1000	-0.1150	4.975×10^{-5}	1.170×10^{-3}	4.965×10^{-5}	1.392×10^{-3}	28.03
a_2	-0.2238	-0.2238	-0.2390	5.564×10^{-5}	1.198×10^{-3}	5.553×10^{-5}	1.427×10^{-3}	25.71
a_3	-0.0844	-0.0843	-0.0983	5.539×10^{-5}	1.165×10^{-3}	5.528×10^{-5}	1.355×10^{-3}	24.52
a_4	-0.0294	-0.0289	-0.0405	5.010×10^{-5}	1.148×10^{-3}	5.029×10^{-5}	1.269×10^{-3}	25.24



(a)

(b)

Fig. 4. True marginal pdf (dashed curve) for driving process of Example 2 and typical estimates of the pdf (solid curves) produced by (a) the algorithm of Zhao *et al.* (20 estimates overlaid) and (b) the EMAX algorithm (20 estimates overlaid).

residual sequence. Thus, one of the most readily observable problems with the approach, as illustrated in Fig. 4(a), is that all of the estimated means of the constituent densities are necessarily distinct, even when the means of the true densities are identical. Fig. 4(a) shows the true marginal pdf for the driving noise as well as typical estimates of this pdf produced by the Zhao algorithm on separate trials. Observe from the figure that, for about half of the trials, the pdf estimate produced by the Zhao algorithm is off center to the positive side of zero, and for the other half, it is off center to the negative side. On each trial, the estimated Gaussian-mixture pdf is dominated by a single component that attempts to model most of the histogram of the residual samples. However, the resulting overall estimate is always off center because the smaller of the two components in the mixture attempts to model the remaining outliers, which are either much greater or much less than zero. In contrast, as shown in Fig. 4(b), the EMAX algorithm produces pdf estimates that better approximate the true driving-noise pdf.

C. Example 3—AR Process with Laplacian Driving Noise

In many applications, we would like to obtain ML estimates for the parameters of an AR system, but the ML problem is ill-posed because the marginal pdf characterizing the driving noise is unknown. In certain cases, however, it may be reasonable to assume that the true marginal pdf is accurately modeled by a Gaussian-mixture pdf, provided that the means, standard deviations, and weighting coefficients defining the mixture are chosen appropriately. In these cases, if we process our observations with the EMAX algorithm, then we might expect the EMAX algorithm to find the mixture parameters that yield a good approximation to the true driving-noise pdf and simultaneously to produce good approximations to the ML estimates for the AR parameters. With the present example, we demonstrate the validity of this approximate approach to the ML estimation problem.

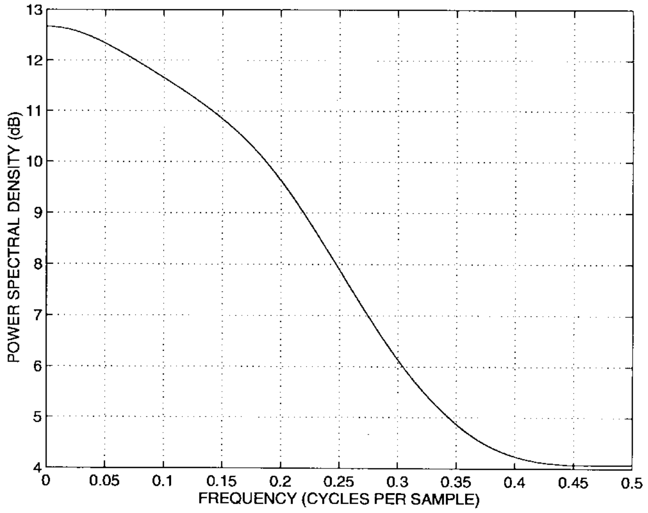


Fig. 5. Power spectral density of fifth-order AR process discussed in Example 3.

In particular, we consider the parameter estimation problem for a fifth-order AR process whose AR coefficients are given by

$$(a_1, a_2, a_3, a_4, a_5) = (1.934, -2.048, 1.072, -0.340, 0.027). \tag{50}$$

The driving noise for this process consists of i.i.d. samples distributed according to a Laplacian pdf defined by

$$f_V(v) = \frac{1}{2\beta} \exp\left\{-\frac{|v|}{\beta}\right\}, \qquad -\infty < v < \infty \tag{51}$$

where the scale parameter β (which is related to the standard deviation σ for this density by $\sigma = \sqrt{2}\beta$) was put at $\beta = 5$. A plot of the power spectral density of this process is shown in Fig. 5.

It will be interesting to compare the performance of the EMAX algorithm with that of the exact ML estimates, which can be computed in this case. It can be shown [2] that if

TABLE III
SAMPLE MEANS AND SAMPLE MEAN SQUARE ERROR (MSE) VALUES FOR PARAMETER ESTIMATES OF EXAMPLE 3. ENTRIES WERE COMPUTED USING RESULTS OF 500 TRIALS FOR i) THE STANDARD FORWARD-BACKWARD LEAST-SQUARES (LS) METHOD, ii) THE EMAX ALGORITHM, AND iii) THE ML ESTIMATION ALGORITHM DEVELOPED BY SCHLOSSMACHER

	True Value	*Sample Mean* (LS)	*Sample Mean* (EMAX)	*Sample Mean* (ML)	*Sample MSE* (LS)	*Sample MSE* (EMAX)	*Sample MSE* (ML)
a_1	1.934	1.9311	1.9323	1.9328	1.0751×10^{-3}	6.3040×10^{-4}	5.7711×10^{-4}
a_2	-2.048	-2.0413	-2.0447	-2.0449	5.1570×10^{-3}	2.8784×10^{-3}	2.7250×10^{-3}
a_3	1.072	1.0647	1.0685	1.0697	8.4001×10^{-3}	4.7769×10^{-3}	4.2887×10^{-3}
a_4	-0.340	-0.3358	-0.3383	-0.3390	4.9714×10^{-3}	2.9190×10^{-3}	2.4228×10^{-3}
a_5	0.027	0.0256	0.0264	0.0269	1.0509×10^{-3}	6.2875×10^{-4}	5.3948×10^{-4}

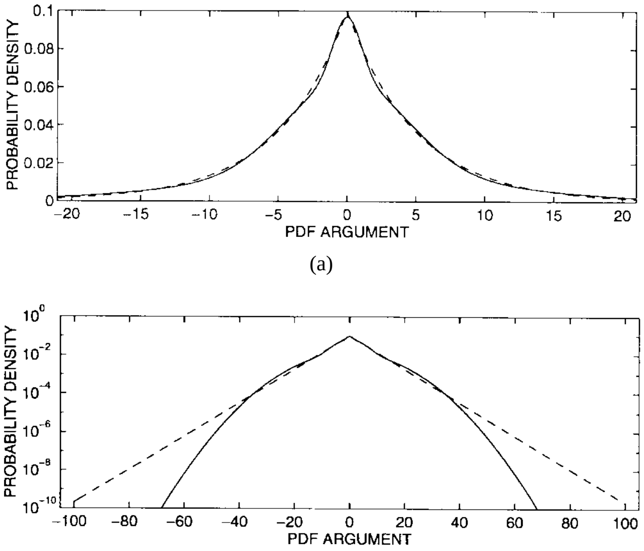
the samples of the driving noise for an AR process are i.i.d. and Laplacian, then the ML estimate for the AR parameter vector $\boldsymbol{a}$ is given by the value of $\boldsymbol{a}'$ that minimizes the sum of absolute residuals $\Sigma_{t=0}^{N-1} |y_t - \boldsymbol{y}_t^T \boldsymbol{a}'|$. An algorithm for finding such a value for $\boldsymbol{a}'$ was proposed by Schlossmacher [15]; this algorithm is based on the method of iteratively reweighted least squares and is therefore easy to implement on a computer.

To find parameter estimates for this problem with the EMAX algorithm, we fixed the number of Gaussian densities in the mixture at $M = 3$ and constrained the means of these constituent densities to be zero. We performed a total of 500 trials. On each trial, a sequence of 1000 data points was generated and processed with the EMAX algorithm. The sample means and sample mean square errors of the parameter estimates produced by the EMAX algorithm are presented in Table III.

A summary of the sample means and sample mean square errors of the AR parameter estimates given by two other algorithms is also shown in Table III: i) the forward-backward least-squares method and ii) the ML algorithm of Schlossmacher. Experimental results shown in Table III confirm our expectation that the ML-based estimator would perform better than the EMAX and least-squares methods since it directly exploits the fact that the driving noise is i.i.d. with a Laplacian distribution. Observe from the table that the ratio of the mean square error of the least-squares estimate to that of the ML estimate ranges approximately from 1.9–2.1. The ratio of the mean square error of the EMAX estimate to that of the ML estimate ranges approximately from 1.1–1.2. Thus, in this case, the EMAX algorithm produces estimates that are much closer to the exact ML estimates than the least-squares estimates.

The superior performance of the EMAX algorithm may be attributed to the ability of its assumed Gaussian-mixture pdf to closely approximate the Laplacian pdf, as is shown for a typical case in Fig. 6(a). It is clear from this figure that the approximation is very good over the region in which most of the samples of the driving noise reside. However, since the number of Gaussian densities in the mixture is finite, an accurate model for the Laplacian density may be obtained only over a finite region of support. Eventually, the tails of the Gaussian-mixture pdf become bounded by a function of the form $k_1 \exp\{-k_2 v^2\}$ for appropriately chosen constants k_1 and k_2. Indeed, Fig. 6(b) reveals this phenomenon with the aid of a log-magnitude scale.

It is interesting to note that although little was initially assumed here about the shape of the driving-noise pdf, the EMAX algorithm is also useful in a commonly encountered variation on this example—namely, in the case where the functional form of the pdf is precisely *known* except for a scale factor. Suppose, for such a case, that ML estimation is extremely difficult (possibly because of the complicated form of the pdf) and, furthermore, that a simple Gaussian assumption leads to unacceptably poor results. In this scenario, the EMAX algorithm may once again provide a convenient way of obtaining a good approximation to the ML solution. To see this, suppose that the true pdf for the driving noise belongs to a parameterized family of pdf's that is invariant with respect to scale (i.e., if the pdf for the random variable X is in the family, then the pdf for cX is also in the family for any positive scalar c). For example, the Laplacian family used in the foregoing computer simulations is scale-invariant, as is the Gaussian family and, hence, the Gaussian-mixture family for any fixed number of mixture components. Since the functional form for the true family of pdf's is assumed known, a good approximation for a particular pdf in this family, using a Gaussian-mixture model, can be designed offline before any data are observed. Once such a Gaussian-mixture approximation is designed, the means and standard



(a)

(b)

Fig. 6. True Laplacian marginal pdf (dashed curve) for driving process of Example 3 and a typical estimate of the pdf (solid curve) produced by the EMAX algorithm, plotted using (a) linear-magnitude scale (with horizontal axis spanning ±3 standard deviations) and (b) log-magnitude scale (with horizontal axis spanning ± 15 standard deviations).

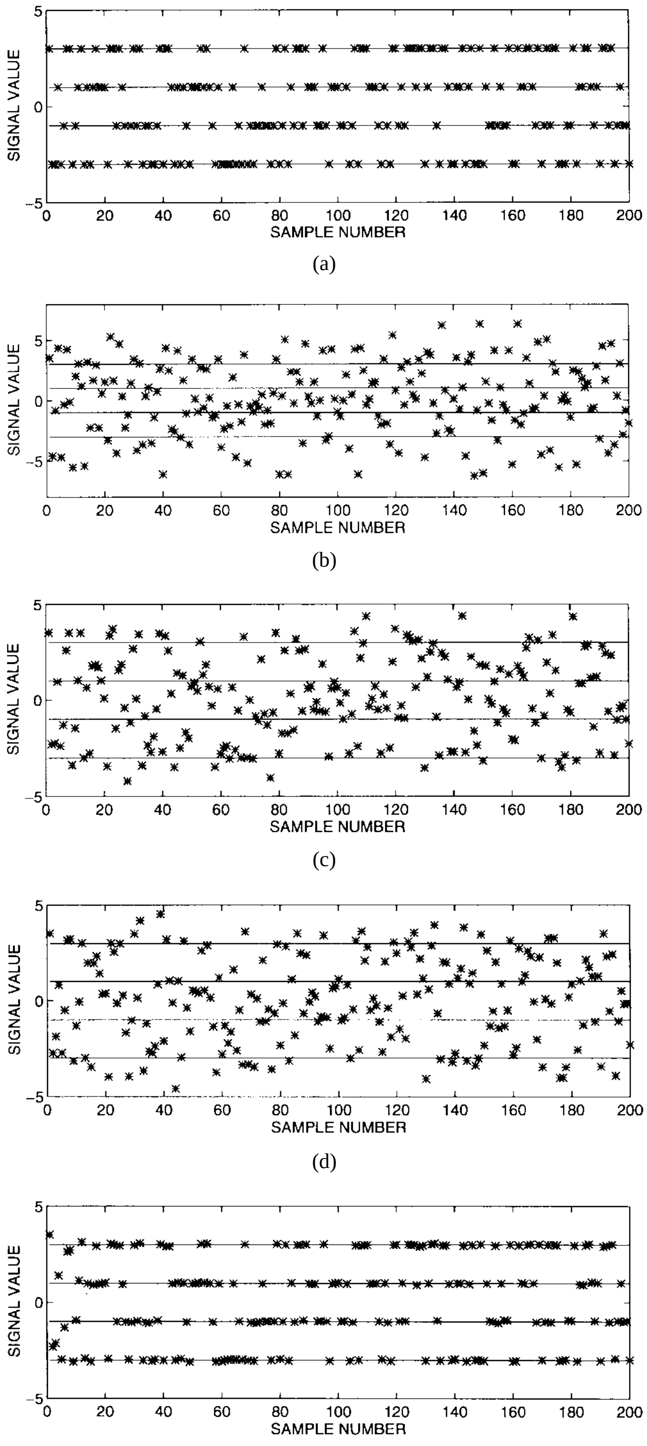
deviations of the mixture components are fixed relative to each other. Moreover, an approximation for any other pdf in the parameterized family can easily be generated from the original approximation by appropriately scaling the means and standard deviations in the Gaussian mixture. The EMAX algorithm can be configured, through a straightforward modification of the original updating formulas, to operate under such a restriction so that it generates approximate joint ML estimates for the AR parameters and for the scale factor associated with the driving-noise pdf.

D. Example 4—Blind Equalization in Digital Communications

Our final example is an application in digital communications that has been adapted from [12]. In this example, we demonstrate that the EMAX algorithm can be used successfully in problems where the primary goal is signal reconstruction, rather than parameter estimation. In particular, we consider a communication system that uses amplitude-shift keying (ASK). In this scheme, the transmitter communicates with the receiver using an L-symbol alphabet $\mathcal{A} = \{A_i\}_{i=1}^{L}$, whose elements we take to be real numbers. To send the kth symbol of a particular message sequence $\{u_t\}$ to the receiver, the transmitter generates a pulse (having fixed shape) and modulates this pulse with the amplitude u_k. The pulse then propagates through the communication medium, which we assume is well modeled by an LTI system. Finally, the receiver processes the waveform with a linear filter to facilitate estimation of u_k.

If this filtered waveform is sampled at a rate of one sample per symbol, then the overall communication system, i.e., the transmitter, the medium, and the receiver, can be represented with an equivalent discrete-time LTI system, which we refer to as the discrete-time channel. In this case, the sampled output is the convolution of the transmitted symbol sequence $\{u_t\}$ and the impulse response $\{h_t\}$ that characterizes the discrete-time channel. If the impulse response $\{h_t\}$ is anything but a shifted and scaled unit impulse, then each sample of the output sequence will contain contributions from more than one input symbol, i.e., there will be intersymbol interference (ISI). If the characteristics of the medium are known, then the discrete-time channel is also known, and the receiver can compensate for the ISI using a linear filter; this technique is known as linear equalization. Often, however, the characteristics of the medium are unknown, and the impulse response of the discrete-time channel must first be estimated in order to compensate for the ISI. One approach for accomplishing this is for the transmitter to send through the medium a training sequence that is known to the receiver. The receiver can then identify the impulse response of the discrete-time channel from the output sequence and apply the corresponding inverse filter. However, if the medium is rapidly changing, then this procedure must be performed frequently, and the effective data rate will be substantially reduced. An alternative approach is to perform blind equalization, i.e., to estimate the impulse response of the discrete-time channel from the output *without* knowing the input, and then apply the appropriate inverse filter.

We consider a scenario in which blind equalization must be performed by the receiver. We assume an ASK modulation scheme that uses the four-symbol alphabet $\mathcal{A} = \{-3, -1, 1, 3\}$. A typical 200-point input sequence to the discrete-time channel, which was generated randomly using the alphabet $\mathcal{A}$, is shown in Fig. 7(a). We assume that the discrete-time channel has a finite impulse response $\{h_t\}$ with



(a)

(b)

(c)

(d)

(e)

Fig. 7. Illustration of channel equalization considered in Example 4. (a) Original symbol sequence. (b) Received sequence. (c) Restored sequence using standard forward-backward least-squares method. (d) Restored sequence using fourth-order cumulant-based Giannakis–Mendel algorithm. (e) Restored sequence using EMAX algorithm assuming four-component Gaussian-mixture pdf.

z-transform

$$H(z) = 1.0 - 0.65z^{-1} + 0.06z^{-2} + 0.41z^{-3}. \tag{52}$$

Fig. 7(b) shows the received sequence, which is the convolution of the input sequence shown in Fig. 7(a) and the impulse response $\{h_t\}$. It is evident from this figure that detection of the input symbols from the received sequence would be difficult without further processing.

Our blind equalization approach consists of channel estimation followed by filtering with the inverse of the estimated channel. We compare three methods for estimating the impulse response of the channel from the output sequence shown in Fig. 7(b):

1) the forward-backward least-squares method;
2) the fourth-order cumulant-based technique of Giannakis and Mendel [5];
3) the EMAX algorithm.

We configured all three algorithms to estimate 18 AR coefficients. Such a configuration assumes that the discrete-time channel inverse may be accurately modeled with a system having 18 zeroes and no poles. We further configured the EMAX algorithm to estimate the means and variances of four constituent Gaussian densities. Fig. 7(c)–(e) shows the restored input sequences generated, respectively, by

1) the least-squares method;
2) the Giannakis–Mendel algorithm;
3) the EMAX algorithm.

It is clear from Fig. 7(c)–(e) that the recovered sequence values produced by the EMAX algorithm are much more tightly distributed around the four true symbol values than either the recovered sequence values produced by the least squares method or those produced by the cumulant-based method. Hence, in this case, we would expect superior detection performance using the EMAX algorithm.

Further discussion of this equalization problem, as well as references to other blind equalization techniques, can be found in the paper by Porat and Friedlander [13].

V. Conclusion

We have presented a general iterative technique known as the EMAX algorithm for estimating the parameters of a non-Gaussian autoregressive random process. In particular, we have restricted our attention to a process that can be represented as the output of an autoregressive LTI system driven by a sequence of i.i.d. random variables having a Gaussian-mixture pdf. Although the likelihood function associated with such a process is typically unbounded in the vicinity of undesirable, degenerate parameter values, we have seen in our numerical examples that good estimates can still be obtained by searching for finite local maxima of the likelihood surface. The goal of the EMAX algorithm is to find such local maxima.

The computations that constitute the EMAX algorithm have an intuitively pleasing form, are easy to implement in computer code, and consume little computer memory. The empirical results presented in Section IV suggest that the EMAX algorithm has at least three distinct advantages over other techniques proposed for similar estimation problems.

1) It produces high-quality estimates since it uses the likelihood function as a guide for finding solutions.
2) It converges reliably to a stationary point of the likelihood function by virtue of being a generalized EM algorithm.
3) It is extremely versatile because the Gaussian-mixture pdf is able to model a wide range of densities very well.

Although the EMAX algorithm is a powerful tool for estimating signal parameters, many issues must still be addressed before the algorithm can be transformed into a complete operational system for performing robust AR signal analysis. For example, a reliable method is required for selecting initial parameter estimates. Recall for the examples presented in Section IV that we adopted an initialization method on the basis of its conceptual and computational simplicity. However, since initial estimates are the key to good performance, we need an initialization that will consistently lead to points of high likelihood after the algorithm has been iterated to convergence. In addition, for some problems (particularly those for which the Gaussian-mixture pdf has many individual components), it would be useful to speed up the convergence of the EMAX algorithm. This might be accomplished by iterating the algorithm until reaching the vicinity of a local maximum and then applying a more efficient method (e.g., the Newton–Raphson technique) to move to the peak. In addition, it would be useful to detect, during the operation of the algorithm, whether a degenerate parameter estimate is being approached so that the algorithm could be restarted elsewhere in the parameter space. Another issue that must be addressed is how to estimate the parameters K (the order of the autoregression) and M (the number of constituent densities in the Gaussian mixture) when these parameters are not given in advance and, moreover, how to assess the effect that incorrectly chosen values for K and M would have on the variance of the other parameter estimates. Finally, we note that in any practical setting, our observations of the signal of interest will be corrupted by additive noise. For example, the digital communications application presented in Section IV is a typical case in which additive noise is unavoidable. Hence, a modification of the EMAX algorithm must be devised for estimating the parameters of an ARGMIX process when noise is present.

The Gaussian-mixture assumption for the driving-noise pdf provides a convenient and general parametric framework for analyzing non-Gaussian AR signals. The EMAX algorithm provides a useful way of exploiting this assumption to obtain high-quality estimates of signal parameters. Further research aimed at enhancing the strengths of the EMAX algorithm in relation to other inherently limited techniques could make the EMAX algorithm a standard technique for solving practical signal processing problems.

References

[1] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *J. R. Statist. Soc.*, Series B, pp. 1–38, 1977.

[2] N. R. Draper and H. Smith, *Applied Regression Analysis*. New York: Wiley, 1950.

[3] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[4] B. S. Everitt and D. J. Hand, *Finite Mixture Distributions*. London, U.K.: Chapman and Hall, 1981.

[5] G. B. Giannakis and J. M. Mendel, "Identification of nonminimum phase systems using higher order statistics," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 360–377, Mar. 1989.

[6] D. G. Luenberger, *Linear and Nonlinear Programming*, 2nd ed. Reading, MA: Addison Wesley, 1984.

[7] S. L. Marple, *Digital Spectral Analysis with Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1987.

[8] G. J. McLachlan and K. E. Basford, *Mixture Models*. New York: Marcel Dekker, 1988.

[9] J. M. Mendel, "Tutorial on higher-order statistics (spectra) in signal processing and system theory: Theoretical results and some applications," *Proc. IEEE*, vol. 79, Mar. 1991.

[10] C. L. Nikias and M. R. Raghuveer, "Bispectrum estimation: A digital signal processing framework," *Proc. IEEE*, vol. 75, pp. 869–891, July 1987.

[11] C. L. Nikias and A. P. Petropulu, *Higher-Order Spectra Analysis: A Nonlinear Signal Processing Framework*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[12] B. Porat, *Digital Processing of Random Signals: Theory and Methods*. Englewood Cliffs, NJ: Prentice-Hall, 1994.

[13] B. Porat and B. Friedlander, "Blind equalization of digital communication channels using high-order moments," *IEEE Trans. Signal Processing*, vol. 39, pp. 522–526, Feb. 1991.

[14] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Rev.*, vol. 26, no. 2, Apr. 1984.

[15] E. J. Schlossmacher, "An interactive technique for absolute deviations curve fitting," *J. Amer. Statist. Assoc.*, vol. 68, no. 344, Dec. 1973.

[16] D. Sengupta and S. Kay, "Efficient estimation of parameters for non-Gaussian autoregressive processes," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, June 1989.

[17] D. M. Titterington, A. F. M. Smith, and U. E. Makov, *Statistical Analysis of Finite Mixture Distributions*. Chichester, U.K.: Wiley, 1985.

[18] C. F. J. Wu, "On the convergence properties of the EM algorithm," *Ann. Statist.*, vol. 11, no. 1, pp. 95–103, 1983.

[19] Y. Zhao, X. Zhuang, and S.-J. Ting, "Gaussian mixture density modeling of non-Gaussian source for autoregressive process," *IEEE Trans. Signal Processing*, vol. 43, Apr. 1995.

Shawn M. Verbout received the B.S. degree in mathematics from Illinois State University, Normal, in 1987 and the S.M. degree in electrical engineering from the Massachusetts Institute of Technology (MIT), Cambridge, in 1994. He is currently pursuing the Ph.D. degree in electrical engineering at MIT.

Since 1987, he has been with MIT Lincoln Laboratory, Lexington, where he has worked primarily on problems in automatic detection and recognition of remotely sensed objects. Since 1992, he has been affiliated with the Digital Signal Processing Group in the Research Laboratory of Electronics, MIT. His current research interests are in the general area of statistical signal processing with particular emphasis on problems involving non-Gaussian signals.

James M. Ooi was born in Singapore in 1970. He received the S.B., S.M., and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology (MIT), Cambridge, in 1992, 1993, and 1997, respectively. From 1995 to 1997, he held an AT&T Bell Laboratories Ph.D. Fellowship.

From 1993 to 1997, he was with the Digital Signal Processing Group at MIT. During the same period, he was a teaching assistant for classes in computer programming, digital signal processing, and statistical signal processing. He has been on the technical staff at the Texas Instruments Computer Science Center, from 1989 to 1992, Orbital Sciences Corporation in 1993, and AT&T/Lucent Technologies Bell Laboratories from 1995 to 1996, where he has researched a variety of topics including computer vision, speech and audio coding, satellite communication, and error-correcting codes. He is currently a consultant with the Mitchell Madison Group, Cambridge, MA, a strategic management consulting firm. His current interests include information theory, fuzzy logic, game theory, and estimation.

Jeffrey T. Ludwig received the S.B. degree in aeronautics and astronautics in 1991, the S.M. degree in electrical engineering and computer science in 1993, and the Ph.D. degree in electrical engineering and computer science in 1997, all from the Massachusetts Institute of Technology (MIT), Cambridge.

He has served as a teaching assistant for graduate and undergraduate classes at MIT in digital signal processing, feedback systems, and control theory, has authored more than a dozen technical publications, and holds one U.S. patent. He has five years experience as a principal engineer with Hughes Electronics Corporation, two years experience as a research assistant at the MIT Lincoln Laboratory, and one year experience as a research associate with the Signal Processing Center of Technology, Lockheed Martin Corporation. He recently joined Delphi Structured Finance Corporation, where he has focused on the development and marketing of leading edge loan portfolio analytics and cash management systems for the securitization industry.

Alan V. Oppenheim (F'77) received the S.B. and S.M. degrees in 1961 and the Sc.D. degree in 1964, all in electrical engineering, from the Massachusetts Institute of Technology (MIT), Cambridge. He was also the recipient of an honorary doctorate from Tel Aviv University, Tel-Aviv, Israel, in 1995.

In 1964, he joined the faculty at MIT, where he is currently the Ford Professor of Engineering and a MacVicar Faculty Fellow. Since 1967, he has also been affiliated with MIT Lincoln Laboratory and since 1977 with the Woods Hole Oceanographic Institution. His research interests are in the general area of signal processing and its applications. He is coauthor of the widely used textbooks *Discrete-Time Signal Processing* and *Signals and Systems*. He is also editor of several advanced books on signal processing.