Nonlinear Maximum Likelihood Estimation of Autoregressive Time Series

L. Todd McWhorter and Louis L. Scharf, *Fellow, IEEE*

***Abstract*—In this paper, we describe an algorithm for finding the *exact, nonlinear*, maximum likelihood (ML) estimators for the parameters of an autoregressive time series. We demonstrate that the ML normal equations can be written as an interdependent set of cubic and quadratic equations in the AR polynomial coefficients. We present an algorithm that algebraically solves this set of nonlinear equations for low-order problems. For high-order problems, we describe iterative algorithms for obtaining a ML solution.**

I. Introduction

In this paper, we derive a new algorithm for computing maximum likelihood (ML) estimators of the parameters that characterize a stationary Gaussian autoregressive time series. The derivation is based on the Gohberg–Semencul formula for the inverse of a Toeplitz matrix. Our key result is a set of equations we have labeled the normal equations of maximum likelihood, to distinguish them from the normal equations of linear prediction. The normal equations of maximum likelihood are at most cubic in the autoregressive parameters, whereas the normal equations of linear prediction are, of course, linear. We present two approaches for solving the nonlinear ML normal equations: an algebraically exact algorithm based on the properties of Sylvester resolvent matrices and approximate solution by iterated map.

Any attempt to summarize the vast literature on autoregressive modeling, or identification of autoregressive time series, would be futile. Nonetheless, by reviewing the main lines of research over the past 70 years, we can establish the context of the results in this paper.

With reference to Table I, we organize work on autoregressive (AR) modeling according to the criterion for identification, and the representation used to describe the AR model. These are the columns and rows of Table I. Beginning in column 1, we classify Burg's algorithm [4] for identifying a sequence of reflection coefficients as a recursive linear prediction (RLP) technique that uses a Levinson recursion for the sequence of approximating AR models. There have been, to date, no other RLP algorithms based on other representations of the AR time series. These classifications account for the first column of the table.

Manuscript received March 11, 1994; revised May 14 1995. This work was supported by Bonneville Power Administration under Contract #DEBI7990BPO7346 and by the Office of Naval Research, Statistics and Probability Branch, under Contract N00014-89-J1070. The associate editor coordinating review of this paper and approving it for publication was Dr. Monique Fargues.

The authors are with Department of Electrical and Computer Engineering, University of Colorado, Boulder, CO 80309 USA.

IEEE Log Number 9415266.

The literature cited in column 2 has the common objective of minimizing prediction error variance and is classified as linear prediction (LP). The work of Yule [21], Walker [20], and Durbin [8] is classified as linear prediction (LP), using a Toeplitz representation for the estimated correlation matrix. Actually, the work of Durbin belongs to two classifications, because it advocates the use of the Levinson recursions for the efficient solution of the normal equations. The work of Morf, *et al.* [15], LeRoux and Gueguen [13], Friedlander *et al.* [9], and Demeure and Scharf [6] is classified as LP, with a Levinson representation for the inverse correlation matrix. This work provides fast algorithms for solving the normal equations of linear prediction when the estimated correlation matrix is close to Toeplitz. These classifications account for the second column of the table.

In column 3, the work of Kay [12] is difficult to classify because it uses two representations for the AR time series. That is, it uses a Gohberg–Semencul characterization of $\mathbf{R}^{-1}$, but it uses a Levinson formula to represent the AR model in the recursive maximization scheme. We classify this work as recursive maximum likelihood (RML), with a Gohberg–Semencul formula for the inverse correlation matrix. The work of Vis and Scharf [19] is classified as RML with a Levinson formula for the representation of $\mathbf{R}^{-1}$ and the order increasing AR models. It clarifies the connection between Kay's work on RML and Burg's work on RLP, and completes the classification of the literature in the third column of the table.

In column 4, the theory of exact maximum likelihood (ML) estimation of AR parameters begins with the work of Schweppe [18], although he provided no algorithms for the maximization of likelihood. Akaike [1] and Ansley [2] did provide such algorithms. This work is classified as ML, based on a Markovian representation for the time series and its correlation sequence.

The work of Morf *et al.* [16] provided a link between Markovian representations and Levinson recursions, leading to the formulas of Dugre *et al.* [7] for computing likelihood. Neither [16] nor [7] contained formulas for maximizing likelihood.

The work of Kailath *et al.* [11], Box and Jenkins [3], and this paper are classified as ML, based on the Gohberg–Semencul formula for the inverse correlation matrix. No algorithms were presented in [11] or [3] for maximizing likelihood. Our contribution is to maximize likelihood by deriving a new set of nonlinear normal equations, based on the Gohberg–Semencul formula, and to present algorithms for solving them. The work of Burg *et al.* [5] is not exact ML because it used ML

TABLE I
CLASSIFICATION OF ALGORITHMS BY CRITERION AND REPRESENTATION OF $\mathbf{R}^{-1}$

Repr. for $\mathbf{R}$, $\mathbf{R}^{-1}$, or $A(z)$	Criterion			
	RLP	LP	RML	ML
Toeplitz only	NA	Yule [21] Walker [20] Durbin [8]	NA	Burg *et al.* [5]
Levinson	Burg [4]	Morf *et al.* [15] LeRoux-Gueguen [13] Friedlander *et al.* [9] Demeure-Scharf [6]	Vis-Scharf [19]	Morf *et al.* [16] Dugre *et al.* [7]
Gohberg-Semencul	NA	NA	Kay [12]	Kailath *et al.* [11] Box-Jenkins [3] McWhorter-Scharf (22)
Markovian	NA	NA	NA	Schweppe [18] Akiake [1] Ansley [2]

to estimate a Toeplitz correlation matrix without assuming a model for the time series. Approximate ML estimates of the AR coefficients are then obtained by solving the normal equations of linear prediction using the estimated correlation matrix.

This paper is organized as follows. In Section II, we use a Gohberg–Semencul formula to derive a novel set of normal equations in the AR coefficients. These normal equations illuminate both the similarities and the differences between linear prediction and exact maximum likelihood. In Section III, we describe an algebraically exact algorithm for solving the nonlinear normal equations of ML. Sections III-A and III-B describe the algorithm for first- and second-order processes. Section III-C extends the algebraically exact algorithm to systems of arbitrary order. Section III-D is concerned with the computational aspects of this algorithm. In Section IV we briefly describe some iterative procedures for solving the normal equations derived in Section II. These iterative algorithms differ from those of Burg and Kay in that we are iteratively solving the exact ML normal equations and not iteratively maximizing an approximation to the true likelihood function.

II. NORMAL EQUATIONS

Let $\mathbf{y} = [y_0\ y_1\ \cdots\ y_{N-1}]^T$ be a vector of data from an autoregressive (AR) time series. That is, assume the time series is synthesized according to the model of Fig. 1. The wide-sense stationary time series $\{u_t\}$ is modeled as a zero-mean white Gaussian noise process of variance σ^2. The pth-order polynomial $A(z) = 1 + a_1 z^{-1} + \cdots + a_p z^{-p}$ is assumed to be monic and minimum phase with real-valued coefficients. These assumptions imply that the snapshot $\mathbf{y}$ is distributed $N[\mathbf{0}, \mathbf{R}]$ where $\mathbf{R} \in R^{N\times N}$ is the symmetric Toeplitz correlation matrix

$$\mathbf{R} = \begin{bmatrix} r_0 & r_1 & \cdots & r_{N-1} \\ r_1 & r_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & r_1 \\ r_{N-1} & \cdots & r_1 & r_0 \end{bmatrix}.$$

We denote any $(p+1)\times(p+1)$ block on the diagonal of $\mathbf{R}$ by $\mathbf{R}_p$. In the remainder of this section we derive the normal

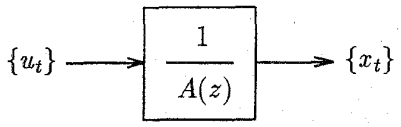


Fig. 1. Synthesis model for an AR time series.

equations for the ML estimators of the input noise variance σ^2 and the AR coefficients $\{a_i\}_1^p$.

Assume that we are given M statistically independent snapshots $\mathbf{Y} = [\mathbf{y}_1\ \mathbf{y}_2\ \cdots\ \mathbf{y}_M]$ of a time series synthesized as in Fig. 1. We assume that each snapshot has N elements. The log-likelihood function for the data can be written as

$$\begin{aligned} L &= -\frac{MN}{2}\ln(2\pi) - \frac{M}{2}\ln(|\mathbf{R}|) - \frac{1}{2}\sum_{i=1}^{M}\mathbf{y}_i^T\mathbf{R}^{-1}\mathbf{y}_i \\ &= -\frac{MN}{2}\ln(2\pi) - \frac{M}{2}\ln(|\mathbf{R}|) - \frac{M}{2}\mathrm{tr}\{\mathbf{R}^{-1}\mathbf{S}\} \end{aligned}$$

where

$$\mathbf{S} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{y}_i\mathbf{y}_i^T$$

is the sample correlation matrix. The correlation matrix, $\mathbf{R}$, is completely described by the AR coefficients and the input noise variance. The Gohberg–Semencul inversion formulas, described in [10], provide one way to represent the correlation matrix in terms of the AR coefficients and noise variance. The key Gohberg–Semencul formula in our derivation is

$$\mathbf{R}^{-1} = \frac{1}{\sigma^2}\mathbf{Q}^{-1} = \frac{1}{\sigma^2}(\mathbf{F}\mathbf{F}^T - \mathbf{G}\mathbf{G}^T) \tag{1}$$

where $\mathbf{F}$ and $\mathbf{G}$ are the $N \times N$ lower triangular Toeplitz matrices

$$\mathbf{F} = \sum_{i=0}^{p} a_i\mathbf{Z}_i = \begin{bmatrix} 1 & & & & \mathbf{0} \\ a_1 & 1 & & & \\ \vdots & \ddots & \ddots & & \\ a_p & & \ddots & \ddots & \\ & \ddots & & \ddots & \ddots \\ \mathbf{0} & & a_p & \cdots & a_1 \quad 1 \end{bmatrix}; \quad a_0 = 1,$$

$$\mathbf{G} = \sum_{i=1}^{p} a_i \mathbf{Z}_{N-i} = \begin{bmatrix} & & & & \mathbf{0} \\ a_p & & & & \\ a_{p-1} & \ddots & & & \\ \vdots & \ddots & \ddots & & \\ a_1 & \cdots & a_{p-1} & a_p & \end{bmatrix}.$$

The $N \times N$ shift matrices $\mathbf{Z}_k$ are defined by

$$\mathbf{Z}_k = \begin{bmatrix} & & & \mathbf{0} \\ 1 & & & \\ & \ddots & & \\ \mathbf{0} & & 1 & \end{bmatrix} = [\delta(k, i-j)]_{i,j}; \quad \mathbf{Z}_0 = \mathbf{I}; \quad \mathbf{Z}_N = \mathbf{0}.$$

That is, $\mathbf{Z}_k$ is zero except for ones on its kth sub-diagonal. Observe that the data dependent term of the log-likelihood function can be written as

$$\begin{aligned} \mathrm{tr}\{\mathbf{R}^{-1}\mathbf{S}\} &= \frac{1}{\sigma^2}\mathrm{tr}\{\mathbf{Q}^{-1}\mathbf{S}\} = \frac{1}{\sigma^2}\mathrm{tr}\{[\mathbf{F}\mathbf{F}^T - \mathbf{G}\mathbf{G}^T]\mathbf{S}\} \\ &= \frac{1}{\sigma^2}\mathrm{tr}\{\mathbf{F}^T\mathbf{S}\mathbf{F}\} - \frac{1}{\sigma^2}\mathrm{tr}\{\mathbf{G}^T\mathbf{S}\mathbf{G}\} \\ &= \frac{1}{\sigma^2}\mathrm{tr}\left[\sum_{i=0}^{p} a_i \mathbf{Z}_i^T\right]\mathbf{S}\left[\sum_{j=0}^{p} a_j \mathbf{Z}_j\right] \\ &\quad - \frac{1}{\sigma^2}\mathrm{tr}\left[\sum_{i=0}^{p} a_i \mathbf{Z}_{N-i}^T\right]\mathbf{S}\left[\sum_{j=0}^{p} a_j \mathbf{Z}_{N-j}\right] \\ &= \frac{1}{\sigma^2}\sum_{i=0}^{p}\sum_{j=0}^{p} a_i a_j [\mathrm{tr}\{\mathbf{Z}_i^T\mathbf{S}\mathbf{Z}_j\} \\ &\quad - \mathrm{tr}\{\mathbf{Z}_{N-i}^T\mathbf{S}\mathbf{Z}_{N-j}\}]. \end{aligned} \tag{2}$$

Define the vector of AR coefficients $\mathbf{a} = [1\ a_1\ \cdots\ a_p]^T$ and the $(p+1) \times (p+1)$ matrix $\hat{\mathbf{R}}_p(\mathbf{Y}) = [\hat{r}_{ij}(\mathbf{Y})]$ where

$$\hat{r}_{ij}(\mathbf{Y}) = \frac{1}{N}\left[\mathrm{tr}\{\mathbf{Z}_i^T\mathbf{S}\mathbf{Z}_j\} - \mathrm{tr}\{\mathbf{Z}_{N-i}^T\mathbf{S}\mathbf{Z}_{N-j}\}\right]. \tag{3}$$

Then, (2), in conjunction with (3), implies

$$\mathrm{tr}\{\mathbf{R}^{-1}\mathbf{S}\} = \frac{N}{\sigma^2}\mathbf{a}^T\hat{\mathbf{R}}_p(\mathbf{Y})\mathbf{a}. \tag{4}$$

For a single snapshot ($M = 1$), the matrix $\hat{\mathbf{R}}_p(\mathbf{Y})$ has the form

$$\hat{\mathbf{R}}_p(\mathbf{Y}) = \frac{1}{N}(\mathbf{Y}_1^T\mathbf{Y}_1 - \mathbf{Y}_2^T\mathbf{Y}_2) \tag{5}$$

where the Hankel matrix $\mathbf{Y}_1$ and the Toeplitz matrix $\mathbf{Y}_2$ are

$$\mathbf{Y}_1 = \begin{bmatrix} y_0 & y_1 & \cdots & y_p \\ y_1 & y_2 & \cdots & y_{p+1} \\ \vdots & \vdots & & \vdots \\ y_{N-p-1} & y_{N-p} & \cdots & y_{N-1} \\ \vdots & \vdots & \iddots & \\ y_{N-2} & y_{N-1} & & \\ y_{N-1} & & & \mathbf{0} \end{bmatrix}$$

$$\mathbf{Y}_2 = \begin{bmatrix} 0 & y_{N-1} & y_{N-2} & \cdots & y_{N-p} \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & y_{N-2} \\ & & & & y_{N-1} \\ & & & & \\ \mathbf{0} & & & & \end{bmatrix}.$$

For multiple snapshots, $\hat{\mathbf{R}}_p(\mathbf{Y})$ can be defined by (3). Equivalently, $\hat{\mathbf{R}}_p(\mathbf{Y})$ can be built by using the M snapshots to construct M matrices with the structure defined in (5) and then forming their average.

Equation (4) has been derived by Kay [12] and is fundamental in the development of the RML algorithm described in his paper. The matrix $\hat{\mathbf{R}}_p(\mathbf{Y})$ is also important in both the computation and intuitive understanding of our algorithm. Accordingly, we now discuss two important properties of $\hat{\mathbf{R}}_p(\mathbf{Y})$.

Let $B(z) = 1 + b_1 z^{-1} + \cdots + b_p z^{-p}$ be any pth-order *minimum phase* monic polynomial and $\mathbf{b} = [1\ b_1\ \cdots\ b_p]^T$ the corresponding vector of polynomial coefficients. The derivation of (4) implies that

$$\mathbf{b}^T\hat{\mathbf{R}}_p(\mathbf{Y})\mathbf{b} \geq 0$$

for all $\mathbf{Y} = [\mathbf{y}_1\ \cdots\ \mathbf{y}_M]$ and all minimum phase polynomials of order less than $p+1$. To demonstrate this, create a nonnegative definite Toeplitz correlation matrix, $\mathbf{R}_B$, by passing white noise through the AR system $1/B(z)$. The derivation of (4) can be duplicated, substituting $B(z)$ for $A(z)$, to arrive at

$$\sum_{i=1}^{M} \mathbf{y}_i^T\mathbf{R}_B^{-1}\mathbf{y}_i = \mathbf{b}^T\hat{\mathbf{R}}_p(\mathbf{Y})\mathbf{b} \geq 0.$$

This property is important because $\hat{\mathbf{R}}_p(\mathbf{Y})$ is not necessarily nonnegative definite [12]. As we will soon show, the maximum likelihood estimate of σ^2 is the quadratic term $\mathbf{a}^T\hat{\mathbf{R}}_p(\mathbf{Y})\mathbf{a}$. The above property indicates that if the estimated polynomial $A(z)$ is minimum phase, then the estimated noise variance will be positive.

Despite the notation, $\hat{\mathbf{R}}_p(\mathbf{Y})$ is not, in general, a good estimator of the $(p+1) \times (p+1)$ correlation matrix $\mathbf{R}_p$. One deficiency of $\hat{\mathbf{R}}_p(\mathbf{Y})$ is that it is not guaranteed to be nonnegative definite. Moreover, we now show that $\hat{\mathbf{R}}_p(\mathbf{Y})$ is a biased estimator of $\mathbf{R}_p$. From (3), it follows that

$$\begin{aligned} E\{\hat{r}_{ij}(\mathbf{Y})\} &= \frac{1}{N}\mathrm{tr}\{\mathbf{Z}_i^T E\{\mathbf{S}\}\mathbf{Z}_j\} - \frac{1}{N}\mathrm{tr}\{\mathbf{Z}_{N-i}^T E\{\mathbf{S}\}\mathbf{Z}_{N-j}\} \\ &= \frac{1}{N}\mathrm{tr}\{\mathbf{Z}_i^T\mathbf{R}\mathbf{Z}_j\} - \frac{1}{N}\mathrm{tr}\{\mathbf{Z}_{N-i}^T\mathbf{R}\mathbf{Z}_{N-j}\}. \end{aligned} \tag{6}$$

After some algebra, we obtain

$$E\{\hat{\mathbf{R}}_p(\mathbf{Y})\} = \mathbf{R}_p - \mathbf{B}$$

where

$$\mathbf{R}_p = \begin{bmatrix} r_0 & r_1 & \cdots & r_p \\ r_1 & r_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & r_1 \\ r_p & \cdots & r_1 & r_0 \end{bmatrix} \tag{7}$$

$$\mathbf{B} = \frac{1}{N}\begin{bmatrix} 0 & r_1 & \cdots & pr_p \\ r_1 & 2r_0 & \cdots & (p+1)r_{p-1} \\ 2r_2 & 3r_1 & \cdots & (p+2)r_{p-2} \\ \vdots & \vdots & & \vdots \\ pr_p & (p+1)r_{p-1} & \cdots & 2pr_0 \end{bmatrix}. \quad (8)$$

These equations indicate that $\hat{\mathbf{R}}_p(\mathbf{Y})$ is a biased estimator of $\mathbf{R}_p$, with bias $\mathbf{B}$. This property is important for gaining an intuitive understanding of our maximum likelihood algorithm. We address this topic in the later stages of this section.

At this point, we have established that the log-likelihood function, ignoring irrelevant constants, is

$$L = -\frac{MN}{2}\ln(\sigma^2) - \frac{M}{2}\ln(|\mathbf{Q}|) - \frac{MN}{2\sigma^2}\mathbf{a}^T\hat{\mathbf{R}}_p(\mathbf{Y})\mathbf{a}.$$

The normal equations of maximum likelihood can be derived by differentiating the Lagrangian

$$\mathcal{L} = -\frac{MN}{2}\ln(\sigma^2) - \frac{M}{2}\ln(|\mathbf{Q}|) - \frac{MN}{2\sigma^2}\mathbf{a}^T\hat{\mathbf{R}}_p(\mathbf{Y})\mathbf{a} - \lambda(\mathbf{a}^T\delta - 1) \quad (9)$$

with respect to the unknown parameters. Here $\delta = [1\ 0\ \cdots\ 0]^T$ is the first column of the $(p+1)\times(p+1)$ identity matrix. Differentiating (9) with respect to σ^2 yields the normal equation

$$-\frac{MN}{2\sigma^2} + \frac{MN}{2\sigma^4}\mathbf{a}^T\hat{\mathbf{R}}_p(\mathbf{Y})\mathbf{a} = 0.$$

Solving this equation for σ^2 produces the maximum likelihood estimate for σ^2

$$\sigma^2 = \mathbf{a}^T\hat{\mathbf{R}}_p(\mathbf{Y})\mathbf{a}. \quad (10)$$

Therefore, if our estimate of $A(z)$ is minimum phase, the maximum likelihood estimate of σ^2 will be nonnegative.

To find the ML estimator of the AR coefficients, first observe that

$$\frac{\partial}{\partial a_i}\ln(|\mathbf{Q}|) = \mathrm{tr}\left\{\mathbf{Q}^{-1}\frac{\partial \mathbf{Q}}{\partial a_i}\right\} = -\mathrm{tr}\left\{\mathbf{Q}\frac{\partial \mathbf{Q}^{-1}}{\partial a_i}\right\}$$

and

$$\frac{\partial \mathbf{Q}^{-1}}{\partial a_i} = \frac{\partial}{\partial a_i}(\mathbf{F}\mathbf{F}^T - \mathbf{G}\mathbf{G}^T) = \mathbf{Z}_i\mathbf{F}^T + \mathbf{F}\mathbf{Z}_i^T - \mathbf{Z}_{N-i}\mathbf{G}^T - \mathbf{G}\mathbf{Z}_{N-i}^T.$$

These results imply

$$\begin{aligned}\frac{\partial}{\partial a_i}\ln(|\mathbf{Q}|) &= -2\mathrm{tr}\{\mathbf{Q}[\mathbf{Z}_i\mathbf{F}^T - \mathbf{Z}_{N-i}\mathbf{G}^T]\} \\ &= -\frac{2}{\sigma^2}\mathrm{tr}\{\mathbf{R}[\mathbf{Z}_i\mathbf{F}^T - \mathbf{Z}_{N-i}\mathbf{G}^T]\} \\ &= -\frac{2}{\sigma^2}\sum_{j=0}^{p} a_j\mathrm{tr}\{\mathbf{R}[\mathbf{Z}_i\mathbf{Z}_j^T - \mathbf{Z}_{N-i}\mathbf{Z}_{N-j}^T]\}.\end{aligned} \quad (11)$$

Define the row vector $\mathbf{c}_i^T = [c_{i0}\ c_{i1}\ \cdots\ c_{ip}]$ and the matrix $\mathbf{C}$ where

$$\begin{aligned}c_{ij} &= \mathrm{tr}\{\mathbf{R}[\mathbf{Z}_i\mathbf{Z}_j^T - \mathbf{Z}_{N-i}\mathbf{Z}_{N-j}^T]\} \\ &= \mathrm{tr}\{\mathbf{Z}_j^T\mathbf{R}\mathbf{Z}_i\} - \mathrm{tr}\{\mathbf{Z}_{N-j}^T\mathbf{R}\mathbf{Z}_{N-i}\}\end{aligned} \quad (12)$$

and

$$\mathbf{C} = \begin{bmatrix} \mathbf{c}_0^T \\ \mathbf{c}_1^T \\ \vdots \\ \mathbf{c}_p^T \end{bmatrix}. \quad (13)$$

The result of (11) and the definitions of (12) and (13) can be used to write

$$\frac{\partial}{\partial a_i}\ln(|\mathbf{Q}|) = -\frac{2}{\sigma^2}\mathbf{c}_i^T\mathbf{a}$$

and

$$\frac{\partial}{\partial \mathbf{a}}\ln(|\mathbf{Q}|) = -\frac{2}{\sigma^2}\mathbf{C}\mathbf{a}.$$

We are now in a position to derive the normal equations for the ML estimators of the AR coefficients. The gradient of the Lagrangian in (9) with respect to the vector $\mathbf{a}$ generates the normal equations

$$\begin{aligned}\frac{M}{\sigma^2}\mathbf{C}\mathbf{a} - \frac{MN}{\sigma^2}\hat{\mathbf{R}}_p(\mathbf{Y})\mathbf{a} &= \lambda\delta \\ \left[\frac{1}{N}\mathbf{C} - \hat{\mathbf{R}}_p(\mathbf{Y})\right]\mathbf{a} &= \frac{\lambda\sigma^2}{MN}\delta.\end{aligned} \quad (14)$$

From (12) and (6) through (8), it can be shown that the $(p+1)\times(p+1)$ matrix $\mathbf{C}$ is equivalent to

$$\mathbf{C} = N(\mathbf{R}_p - \mathbf{B})$$

where $\mathbf{R}_p$ and $\mathbf{B}$ are defined in (7) and (8). The ML normal equations can now be written as

$$[\mathbf{R}_p - \mathbf{B} - \hat{\mathbf{R}}_p(\mathbf{Y})]\mathbf{a} = \frac{\lambda\sigma^2}{MN}\delta. \quad (15)$$

Recall that for an AR time series, $\mathbf{R}_p\mathbf{a} = \sigma^2\delta$. Therefore, (15) can be reduced to

$$[\hat{\mathbf{R}}_p(\mathbf{Y}) + \mathbf{B}]\mathbf{a} = \sigma^2\left(1 - \frac{\lambda}{MN}\right)\delta. \quad (16)$$

The important nonlinear term in these normal equations is

$$\mathbf{B}\mathbf{a} = \frac{1}{N}\begin{bmatrix} 0 & r_1 & \cdots & pr_p \\ r_1 & 2r_0 & \cdots & (p+1)r_{p-1} \\ \vdots & \vdots & & \vdots \\ pr_p & (p+1)r_{p-1} & \cdots & 2pr_0 \end{bmatrix}\mathbf{a}.$$

To simplify this expression, note that the bias matrix has the equivalent representation

$$\mathbf{B} = \frac{1}{N}\begin{bmatrix} 0 & & & 0 \\ & 1 & & \\ & & \ddots & \\ 0 & & & p \end{bmatrix}\mathbf{R}_p + \frac{1}{N}\mathbf{R}_p\begin{bmatrix} 0 & & & 0 \\ & 1 & & \\ & & \ddots & \\ 0 & & & p \end{bmatrix}.$$

Define $\mathbf{D} = \mathrm{diag}\{0, 1, 2, \cdots, p\}$. Again, the relation $\mathbf{R}_p\mathbf{a} = \sigma^2\delta$ can be invoked to establish

$$\begin{aligned}\mathbf{B}\mathbf{a} &= \frac{1}{N}(\mathbf{D}\mathbf{R}_p + \mathbf{R}_p\mathbf{D})\mathbf{a} \\ &= \frac{1}{N}\mathbf{D}\delta\sigma^2 + \frac{1}{N}\mathbf{R}_p\mathbf{D}\mathbf{a} \\ &= \frac{1}{N}\mathbf{R}_p\mathbf{a}_d\end{aligned}$$

where $\mathbf{a}_d = \mathbf{D}\mathbf{a} = [0\ a_1\ 2a_2\ \cdots\ pa_p]^T$. The normal equations can now be reduced to

$$\hat{\mathbf{R}}_p(\mathbf{Y})\mathbf{a} + \frac{1}{N}\mathbf{R}_p\mathbf{a}_d = \sigma^2\left(1 - \frac{\lambda}{MN}\right)\delta. \tag{17}$$

The value of the Lagrange multiplier is easily determined from (17)

$$\mathbf{a}^T\hat{\mathbf{R}}_p(\mathbf{Y})\mathbf{a} + \frac{1}{N}\mathbf{a}^T\mathbf{R}_p\mathbf{a}_d = \sigma^2\left(1 - \frac{\lambda}{MN}\right)$$
$$\sigma^2 + \frac{\sigma^2}{N}\delta^T\mathbf{a}_d = \sigma^2\left(1 - \frac{\lambda}{MN}\right)$$
$$\sigma^2 = \sigma^2\left(1 - \frac{\lambda}{MN}\right) \quad \Rightarrow \lambda = 0.$$

The ML normal equations are therefore

$$[\hat{\mathbf{R}}_p(\mathbf{Y}) + \mathbf{B}]\mathbf{a} = \sigma^2\delta \tag{18}$$

or

$$\hat{\mathbf{R}}_p(\mathbf{Y})\mathbf{a} + \frac{1}{N}\mathbf{R}_p\mathbf{a}_d = \sigma^2\delta. \tag{19}$$

These forms of the ML normal equations provide an intuitive understanding of exact maximum likelihood estimation. Note that $E\{\hat{\mathbf{R}}_p(\mathbf{Y}) + \mathbf{B}\} = \mathbf{R}_p$. Thus, in solving the maximum likelihood problem, we are trying to estimate a set of AR coefficients that generate a bias matrix $\mathbf{B}$ to ameliorate the deficiencies of $\hat{\mathbf{R}}_p(\mathbf{Y})$ and satisfy the normal equations as well.

Equation (18) also provides an intuitive connection between the theory of exact maximum likelihood and the least squares theory of linear prediction. In the least squares theory, the normal equations are $\hat{\mathbf{R}}_{\mathrm{LP}}\mathbf{a} = \sigma^2\delta$ where $\hat{\mathbf{R}}_{\mathrm{LP}}$ is an estimate of the correlation matrix $\mathbf{R}_p$. Thus, linear prediction and exact maximum likelihood share a common structure in their respective normal equations. Linear prediction builds a "reasonable" quadratic estimate of the correlation matrix solely from the data and then finds the optimal whitening polynomial. Whereas exact maximum likelihood builds a quadratic, but deficient, estimate of the correlation matrix and *simultaneously* tries to offset the deficiencies and whiten the "corrected" estimate of the correlation matrix.

In the remainder of this section we manipulate the ML normal equations into a form that makes them amenable to either exact or iterative solution. From (19) we write

$$\hat{\mathbf{R}}_p(\mathbf{Y})\mathbf{a} + \frac{1}{N}\sigma^2\mathbf{Q}_p\mathbf{a}_d = \sigma^2\delta$$

or

$$\mathbf{Q}_p^{-1}\hat{\mathbf{R}}_p(\mathbf{Y})\mathbf{a} + \frac{1}{N}\mathbf{a}_d\sigma^2 = \sigma^2\mathbf{Q}_p^{-1}\delta = \mathbf{a}\sigma^2. \tag{20}$$

The matrix $\mathbf{Q}_p$ is the $(p+1)\times(p+1)$ northwest block of the matrix $\mathbf{Q}$ defined in the Gohberg–Semencul formula of (1). It is simply the normalized representation for the Toeplitz matrix $\mathbf{R}_p$. Recall that our maximum likelihood estimate of σ^2 is $\mathbf{a}^T\hat{\mathbf{R}}_p(\mathbf{Y})\mathbf{a}$. Therefore (20) is equivalent to

$$\left[\mathbf{Q}_p^{-1} - \mathbf{a}\mathbf{a}^T + \frac{1}{N}\mathbf{a}_d\mathbf{a}^T\right]\hat{\mathbf{R}}_p(\mathbf{Y})\mathbf{a} = \mathbf{0}. \tag{21}$$

If we define the $(p+1)\times(p+1)$ analogs of $\mathbf{F}$ and $\mathbf{G}$ we arrive at

$$\left[\mathbf{F}_p\mathbf{F}_p^T - \mathbf{G}_p\mathbf{G}_p^T - \mathbf{a}\mathbf{a}^T + \frac{1}{N}\mathbf{a}_d\mathbf{a}^T\right]\hat{\mathbf{R}}_p(\mathbf{Y})\mathbf{a} = \mathbf{0}. \tag{22}$$

The first element of this vector equation is satisfied for all $\mathbf{a}$ and $\hat{\mathbf{R}}_p(\mathbf{Y})$. That is, the first row of $\mathbf{Q}_p^{-1} - \mathbf{a}\mathbf{a}^T + 1/N\mathbf{a}_d\mathbf{a}^T$ is zero for all pth-order polynomials $A(z)$. We are left with p equations that must be solved for the p AR coefficients $\{a_i\}_1^p$. We will show in the next section that the normal equations can be written as an interdependent set of cubic and quadratic equations in the AR coefficients. We also present algorithms to solve this set of equations.

III. Exact Algorithms

In this section, we present an algorithm for solving the normal equations of (22). This algorithm is said to be *exact* because we characterize the solutions for the AR coefficients as roots of polynomials. In practice there will be errors associated with any root finding algorithm so the results will be exact only in an algebraic sense. We introduce the algorithm for first- and second-order problems and then describe how it can be generalized for higher-order systems.

A. First-Order Example

Assume that the polynomial $A(z) = 1 + a_1z^{-1}$ is restricted to be first-order. For the first-order case $\mathbf{a}_d = [0\ \ a_1]^T$ and

$$\mathbf{F}_p = \begin{bmatrix} 1 & 0 \\ a_1 & 1 \end{bmatrix}; \qquad \mathbf{G}_p = \begin{bmatrix} 0 & 0 \\ a_1 & 0 \end{bmatrix}.$$

The normal equations of (22) are then

$$\left[\mathbf{F}_p\mathbf{F}_p^T - \mathbf{G}_p\mathbf{G}_p^T - \mathbf{a}\mathbf{a}^T + \frac{1}{N}\mathbf{a}_d\mathbf{a}^T\right]\hat{\mathbf{R}}_p(\mathbf{Y})\mathbf{a} = \mathbf{0}$$
$$\begin{bmatrix} 0 & 0 \\ \frac{a_1}{N} & \left(1 - a_1^2\left(\frac{N-1}{N}\right)\right) \end{bmatrix}\hat{\mathbf{R}}_p(\mathbf{Y})\mathbf{a} = \mathbf{0}. \tag{23}$$

We are left with the relevant equation

$$\left[\frac{a_1}{N} \quad \left(1 - a_1^2\left(\frac{N-1}{N}\right)\right)\right]\hat{\mathbf{R}}_p(\mathbf{Y})\mathbf{a} = 0.$$

If we let $\hat{r}_{ij}$ denote the ijth element of $\hat{\mathbf{R}}_p(\mathbf{Y})$, we have the equivalent polynomial expression

$$p(x)|_{x=a_1} = x^3 + x^2\left(\frac{(N-2)\hat{r}_{01}}{(N-1)\hat{r}_{11}}\right) - x\left(\frac{N\hat{r}_{11} + \hat{r}_{00}}{(N-1)\hat{r}_{11}}\right) - \frac{N\hat{r}_{01}}{(N-1)\hat{r}_{11}}\Bigg|_{x=a_1} = 0. \tag{24}$$

Kay [12] derived a method to *approximately* maximize likelihood by estimating a series of reflection coefficients. For AR processes of order two or higher, the method is an approximation to exact maximum likelihood. However for a first-order process, Kay's method should produce the exact ML estimate of a_1. The result of (24) corresponds to the first-order normal equation introduced by Kay. The polynomial of (24) has either one real and two complex conjugate solutions or

three real solutions. Kay has demonstrated that in either case there will be at least one real solution satisfying $|a_1| < 1$. This root is the ML estimate of a_1.

B. Second-Order Example

In this subsection we assume that the polynomial $A(z) = 1 + a_1 z^{-1} + a_2 z^{-2}$ is restricted to be second-order. Under this assumption $\mathbf{a}_d = [0 \; a_1 \; 2a_2]^T$ and

$$\mathbf{F}_p = \begin{bmatrix} 1 & 0 & 0 \\ a_1 & 1 & 0 \\ a_2 & a_1 & 1 \end{bmatrix}; \quad \mathbf{G}_p = \begin{bmatrix} 0 & 0 & 0 \\ a_2 & 0 & 0 \\ a_1 & a_2 & 0 \end{bmatrix}.$$

The relevant normal equations can be written as

$$\begin{bmatrix} \frac{a_1}{N} & 1 - a_2^2 + \frac{a_1^2}{N} & a_1 - a_1 a_2 \left(\frac{N-1}{N}\right) \\ \frac{2a_2}{N} & a_1 - a_1 a_2 \left(\frac{N-2}{N}\right) & 1 - a_2^2 \left(\frac{N-2}{N}\right) \end{bmatrix} \hat{\mathbf{R}}_p(\mathbf{Y})\mathbf{a} = \mathbf{0}.$$

We can also express these equations as

$$\begin{aligned} p_1(x)|_{x=a_1} &= x^3\alpha_3(a_2) + x^2\alpha_2(a_2) + x\alpha_1(a_2) + \alpha_0(a_2)|_{x=a_1} \\ &= 0 \qquad (25) \\ p_2(x)|_{x=a_1} &= x^2\beta_2(a_2) + x\beta_1(a_2) + \beta_0(a_2)|_{x=a_1} = 0 \qquad (26) \end{aligned}$$

where the coefficients $\{\alpha_i\}$ and $\{\beta_i\}$ are functions of a_2 and the observed data $\hat{\mathbf{R}}_p(\mathbf{Y})$. To find the maximum likelihood solution we must solve this interdependent set of equations. One can interpret (25) and (26) as two polynomials $p_1(x)$ and $p_2(x)$, which share a root at $x = a_1$. It is this interpretation that indicates that the properties of the Sylvester resolvent matrix might be applicable in the solution of the normal equations. We first describe the construction and properties of the Sylvester resolvent matrix before solving (25) and (26).

Let $a(x)$ and $b(x)$ be the polynomials

$$\begin{aligned} a(x) &= \sum_{i=0}^{n} a_i x^i = a_n \prod_{i=1}^{n} (x - \alpha_i) \quad a_n \neq 0 \\ b(x) &= \sum_{i=0}^{m} b_i x^i = b_m \prod_{i=1}^{m} (x - \beta_i) \quad b_m \neq 0 \end{aligned}$$

and define the vector $\psi_k(x) = [1 \; x \; x^2 \; \cdots \; x^{k-1}]^T$. The Sylvester resolvent matrix, M, can be implicitly defined [17] by

$$\mathbf{M}\psi_{n+m}(x) = \begin{bmatrix} a(x)\psi_m(x) \\ b(x)\psi_n(x) \end{bmatrix}$$

or explicitly by

$$\mathbf{M} = \begin{bmatrix} a_0 & \cdots & a_n & & & \\ & a_0 & \cdots & a_n & & \mathbf{0} \\ \mathbf{0} & & \ddots & & \ddots & \\ & & & a_0 & \cdots & a_n \\ b_0 & \cdots & b_m & & & \\ & b_0 & \cdots & b_m & & \mathbf{0} \\ \mathbf{0} & & \ddots & & \ddots & \\ & & & b_0 & \cdots & b_m \end{bmatrix}.$$

The Sylvester resolvent matrix is also called a resultant matrix by Heinig and Rost [10]. It can be shown that the determinant of the Sylvester resolvent matrix is

$$\det(\mathbf{M}) = b_m^n \prod_{k=1}^{m} a(\beta_k).$$

Thus, $\det(\mathbf{M}) = 0$ if and only if the polynomials $a(x)$ and $b(x)$ share at least one root. It is this property that we will exploit in the algorithm that follows.

The initial step in the algorithm for finding the ML estimates of a_1 and a_2 is to construct the Sylvester resolvent matrix for the polynomials defined in (25) and (26)

$$\mathbf{M} = \begin{bmatrix} \alpha_0(y) & \alpha_1(y) & \alpha_2(y) & \alpha_3(y) & 0 \\ 0 & \alpha_0(y) & \alpha_1(y) & \alpha_2(y) & \alpha_3(y) \\ \beta_0(y) & \beta_1(y) & \beta_2(y) & 0 & 0 \\ 0 & \beta_0(y) & \beta_1(y) & \beta_2(y) & 0 \\ 0 & 0 & \beta_0(y) & \beta_1(y) & \beta_2(y) \end{bmatrix}_{y=a_2}.$$

To reiterate, the normal equations indicate that the polynomials $p_1(x)$ and $p_2(x)$ share a common root at $x = a_1$, provided the coefficients are evaluated at the maximum likelihood solution for a_2. Therefore, the Sylvester resolvent matrix for these two polynomials must be singular at the ML solution for a_2. This fact can be exploited to obtain the following polynomial in a_2

$$p_3(y)|_{y=a_2} = \det(\mathbf{M}) = \sum_{i=0}^{9} \gamma_i y^i \Big|_{y=a_2} = 0. \qquad (27)$$

The coefficients $\{\gamma_i\}$ of the polynomial $p_3(y)$ are functions only of the data. In Section III-D we discuss some methods for computing these coefficients. Any potential ML solution for a_2 must be a root of the ninth-order polynomial in (27). The algorithm for the second-order case is then:

Exact Algorithm for a Second-Order AR Process

1) Form the matrix $\hat{\mathbf{R}}_p(\mathbf{Y})$ using the time series data. Note that the normal equations for the AR coefficients are invariant to scaling of this matrix.
2) Compute the coefficients $\{\alpha_i(y)\}_0^3$ and $\{\beta_i(y)\}_0^2$ of the polynomials $p_1(x)$ and $p_2(x)$ using (22).
3) Compute the polynomial $p_3(y)$ using (27) or the techniques outlined in Section III-D.
4) Find all *real* roots of $p_3(y)$ with magnitude less than one. These are the potential ML solutions, $\{a_2^{(i)}\}$, for the AR coefficient a_2.
5) Substitute all potential ML solutions for a_2 into (25) or (26). The *real* roots of these equations are potential ML solutions for a_1. To summarize, at this point we have the potential ML solution pairs $(a_1^{(i)}, a_2^{(i)})$ where both coefficients are real and $|a_2^{(i)}| < 1$.
6) Discard all potential solution pairs that do not produce a minimum phase AR polynomial $A(z) = 1 + a_1^{(i)} z^{-1} + a_2^{(i)} z^{-2}$ and/or that do not satisfy *both* normal (25) and (26). Usually only one pair of solutions will remain and this is the ML solution for the AR coefficients. In the numerical trials we have conducted, we have always

obtained at least one minimum phase solution to the ML normal equations. On occasion, if the number of data points is small ($N = 4, 5$), we have obtained multiple, minimum phase, solutions to the ML normal equations. In these cases, the ML solutions differ by only a small amount. It may be that, theoretically, these solutions are identical, but due to numerical inaccuracy we obtain multiple, approximately equal, solutions.

This concludes our discussion of the exact ML algorithm for the second-order example. In the next section we generalize this algorithm for systems with arbitrary order.

C. Exact Algorithm for Higher-Order AR Processes

The general procedure for solving the ML normal equations is a simple extension of the techniques presented in the previous section. Recall that the variables were separated by relying on the fact that two polynomials shared a common root. The same idea applies in the general procedure. In the following we present the generalized algorithm for a third-order system. The algorithm for a general pth-order system is merely an extension of the procedure we present below.

For a third-order system, use the normal equations of (22) to generate the three polynomials

$$p_1(a_2) = a_2^3\alpha_3 + a_2^2\alpha_2 + a_2\alpha_1 + \alpha_0 = 0$$
$$p_2(a_2) = a_2^2\beta_2 + a_2\beta_1 + \beta_0 = 0$$
$$p_3(a_2) = a_2^2\gamma_2 + a_2\gamma_1 + \gamma_0 = 0$$

where the coefficients $\{\alpha_i\}$, $\{\beta_i\}$, and $\{\gamma_i\}$ are functions of the AR coefficients a_1 and a_3 and the data. The formulas for these coefficients can be derived from (22). Now, form two Sylvester resolvent matrices

$$\mathbf{M}_1 = \begin{bmatrix} \alpha_0 & \alpha_1 & \alpha_2 & \alpha_3 & 0 \\ 0 & \alpha_0 & \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_0 & \beta_1 & \beta_2 & 0 & 0 \\ 0 & \beta_0 & \beta_1 & \beta_2 & 0 \\ 0 & 0 & \beta_0 & \beta_1 & \beta_2 \end{bmatrix}$$
$$\mathbf{M}_2 = \begin{bmatrix} \alpha_0 & \alpha_1 & \alpha_2 & \alpha_3 & 0 \\ 0 & \alpha_0 & \alpha_1 & \alpha_2 & \alpha_3 \\ \gamma_0 & \gamma_1 & \gamma_2 & 0 & 0 \\ 0 & \gamma_0 & \gamma_1 & \gamma_2 & 0 \\ 0 & 0 & \gamma_0 & \gamma_1 & \gamma_2 \end{bmatrix}. \quad (28)$$

The normal equations imply that the polynomials $p_1(x)$ and $p_2(x)$ share a common root at a_2. The same assertion is true for $p_1(x)$ and $p_3(x)$. The properties of the Sylvester resolvent matrix can then be used to obtain the two polynomials

$$p_4(a_1) = \det(\mathbf{M}_1) = a_1^q\kappa_q + \cdots + a_1\kappa_1 + \kappa_0 = 0 \quad (29)$$
$$p_5(a_1) = \det(\mathbf{M}_2) = a_1^r\xi_r + \cdots + a_1\xi_1 + \xi_0 = 0. \quad (30)$$

The coefficients $\{\kappa_i\}$ and $\{\xi_i\}$ are now only functions of a_3 and the data. These two equations indicate that the polynomials $p_4(x)$ and $p_5(x)$ share a root at a_1. Therefore the Sylvester resolvent matrix

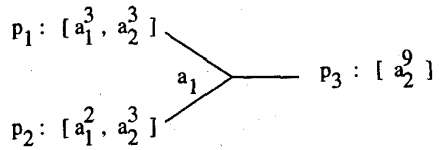


Fig. 2. Resultant polynomial orders for second-order system.

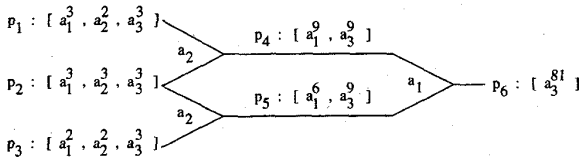


Fig. 3. Resultant polynomial orders for third-order system.

$$\mathbf{M}_3 = \begin{bmatrix} \kappa_0 & \cdots & \kappa_q & & & \\ & \kappa_0 & \cdots & \kappa_q & & 0 \\ 0 & & \ddots & & \ddots & \\ & & & \kappa_0 & \cdots & \kappa_q \\ \xi_0 & \cdots & \xi_r & & & \\ & \xi_0 & \cdots & \xi_r & & 0 \\ 0 & & \ddots & & \ddots & \\ & & & \xi_0 & \cdots & \xi_r \end{bmatrix}$$

must be singular. This property can be exploited to determine our last equation

$$p_6(a_3) = \det(\mathbf{M}_3) = a_3^s\lambda_s + \cdots + a_3\lambda_1 + \lambda_0 = 0 \quad (31)$$

where now the coefficients of this polynomial, $\{\lambda_i\}$, depend exclusively on the data. The AR coefficient a_3 must be a root of the polynomial $p_6(x)$. The procedure is then to find all roots of $p_6(x)$ that are real and have magnitude less than one. All roots that satisfy these conditions are potential solutions for a_3. Potential solutions for a_1 are the real roots of either $p_4(x)$ or $p_5(x)$ with the coefficients of these polynomials formed from the data and the potential solutions for a_3. The solutions for a_2 can be found in a similar fashion from the polynomials $p_1(x)$, $p_2(x)$, or $p_3(x)$. This procedure generates a finite number of potential solution sets for the AR coefficients. The final step of the procedure is to eliminate all sets of solutions that do not generate minimum phase $A(z)$ and/or do not satisfy all three normal equations. It is straightforward to extend this procedure for systems of arbitrary order. In the following section we describe procedures for constructing the coefficients of the polynomials generated by this algorithm.

D. Computational Aspects

The algorithm described in the preceding section requires the computation of a resultant or equivalently the determinant of a Sylvester resolvent matrix. This calculation represents the majority of the computational burden associated with this algorithm. In this section, we discuss some techniques for performing this computation.

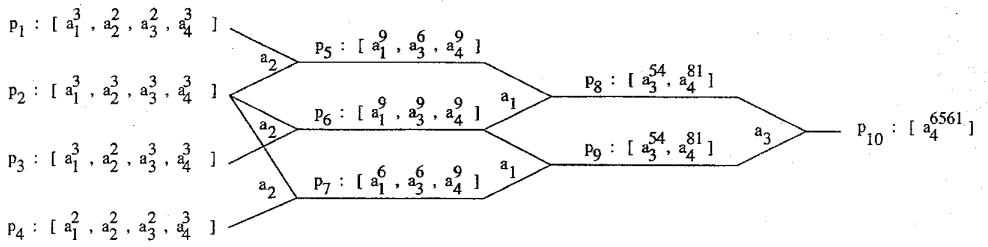


Fig. 4. Resultant polynomial orders for fourth-order system.

The highest degree of computational flexibility is maintained if we obtain the coefficients of the resultant polynomials assuming that the elements of $\hat{\mathbf{R}}_p(\mathbf{Y})$ are variables. In this case the resultant computation has to be performed only once (offline). The data are then used to compute $\hat{\mathbf{R}}_p(\mathbf{Y})$ and the elements of this matrix can be used to directly form the coefficients of the polynomials. For example, in the second-order case, the coefficients of the ninth-order polynomial in (27) depend exclusively on the data through $\hat{\mathbf{R}}_p(\mathbf{Y})$. The resultant operation, which forms this polynomial, can be computed once to obtain maps from $\hat{\mathbf{R}}_p(\mathbf{Y})$ to the coefficients $\{\gamma_i\}_0^9$. The resultant computation in this case usually requires access to a symbolic math software package. However, it has been our experience that these packages are viable for computing these resultants only if the dimensions of the two polynomials are relatively small. Assuming $\hat{\mathbf{R}}_p(\mathbf{Y})$ variable, the symbolic math package we used was easily able to compute the resultants of (27) (for the second-order case) and the resultants of (29) and (30) (third-order case). However, symbolically computing the resultant of (31) was beyond the capabilities of our computer and/or software package. We therefore use this technique to compute the resultants for polynomials of small degree and use the method described below for computing the resultants of polynomials with larger degree.

In this section, we assume that the data have been used to compute numerical values for the elements of $\hat{\mathbf{R}}_p(\mathbf{Y})$. The technique described in this section is easily understood by way of example. Consider the polynomial of (27). If this polynomial was evaluated at 10 different values of y and we knew the result, then the coefficients of the polynomials could be easily found by solving a linear equation. But we know the result of the polynomial evaluated at some number $y = y_i$. It is simply the determinant of the corresponding Sylvester resolvent matrix $\mathbf{M}$, where the elements of $\mathbf{M}$ are evaluated at $y = y_i$. Note that there is no symbolic computation involved in this determinant. The coefficients $\{\gamma_i\}_0^9$ can be obtained by solving the following linear equation

$$\begin{bmatrix} 1 & y_0 & y_0^2 & \cdots & y_0^9 \\ 1 & y_1 & y_1^2 & \cdots & y_1^9 \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & y_9 & y_9^2 & \cdots & y_9^9 \end{bmatrix} \begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \vdots \\ \gamma_9 \end{bmatrix} = \begin{bmatrix} \det\{\mathbf{M}(y_0)\} \\ \det\{\mathbf{M}(y_1)\} \\ \vdots \\ \det\{\mathbf{M}(y_9)\} \end{bmatrix} = \mathbf{c}. \tag{32}$$

Here, $\mathbf{M}(y_i)$ denotes the matrix $\mathbf{M}$ with its elements formed using $y = y_i$. If the $\{y_i\}_0^9$ are chosen to be roots of unity, then (32) can be solved efficiently using an FFT algorithm.

This technique can be extended to higher-order problems. Consider the polynomial of (29) that arose in the third-order case. The coefficients $\{\kappa_i\}_0^q$ are functions of the data and a_3. In fact we can write this polynomial as

$$p_4(a_1, a_3) = \det\{\mathbf{M}_1(a_1, a_3)\} = [1 \; a_1 \; \cdots \; a_1^q]\mathbf{\Gamma} \begin{bmatrix} 1 \\ a_3 \\ \vdots \\ a_3^m \end{bmatrix}$$

where the matrix $\mathbf{\Gamma}$ depends only on the data $\hat{\mathbf{R}}_p(\mathbf{y})$. Let a_1 and a_3 be variable to obtain

$$\det\{\mathbf{M}_1(x_i, y_j)\} = [1 \; x_i \; \cdots \; x_i^q]\mathbf{\Gamma} \begin{bmatrix} 1 \\ y_j \\ \vdots \\ y_j^m \end{bmatrix}$$

or equivalently

$$\mathbf{\Phi} = [\det\{\mathbf{M}_1(x_i, y_j)\}]_{ij} = \mathbf{V}_x^T \mathbf{\Gamma} \mathbf{V}_y$$

where

$$\mathbf{V}_x = \begin{bmatrix} 1 & \cdots & 1 \\ x_0 & & x_q \\ \vdots & & \vdots \\ x_0^q & \cdots & x_q^q \end{bmatrix} \qquad \mathbf{V}_y = \begin{bmatrix} 1 & \cdots & 1 \\ y_0 & & y_m \\ \vdots & & \vdots \\ y_0^m & \cdots & y_m^m \end{bmatrix}.$$

Then, $\mathbf{\Gamma}$ satisfies

$$\mathbf{\Gamma} = \mathbf{V}_x^{-T} \mathbf{\Phi} \mathbf{V}_y^{-1}.$$

Again, this computation can be done efficiently by choosing the $\{x_i\}$ and $\{y_i\}$ to be roots of unity. This technique can be extended to higher-order problems by careful repeated application of the Kronecker product operator. To use this method it is necessary to know the orders of each coefficient in the resultant polynomial. In Figs. 2–4, we list the order of resultant polynomials for systems of order two through four. In these figures the superscript on the AR coefficients indicates the largest degree of the coefficient. It may appear that this algorithm is intractable for systems of order four because the final polynomial (in a_4) has such large degree. However, we know that $|a_4| < 1$ and that a_4 must be real. This additional information can make it possible to use this technique for systems of order four. That is, it is only necessary to search for real roots of p_{10} with modulus less than one. One can also use an initial estimate of a_4 (say from LP) and a Newton–Raphson

procedure to obtain the ML value for a_4. In this case it is not necessary to root the polynomial p_{10}.

This algorithm relies on the computation of the determinant of a Sylvester resolvent matrix (for example to form $\mathbf{c}$ or $\mathbf{\Phi}$). In the following we summarize a recursive procedure derived in [14] for computing these determinants. Let $a(x) = \sum a_i x^i$ be a polynomial of degree n. Let $b(x) = \sum b_i x^i$ be a polynomial of degree m, and assume that $n \geq m$. Let $\mathbf{M}$ denote the Sylvester resolvent matrix for $a(x)$ and $b(x)$ and define the resultant $R[a(x), b(x)] = \det\{\mathbf{M}\}$. Then it can be shown that $R[a(x), b(x)] = b_m R[c(x), b(x)]$ where the polynomial $c(x) = a(x) - x^{(n-m)}(a_n/b_m)b(x)$ now has, at most, degree $n-1$. This procedure can be applied recursively to compute the determinants.

IV. Iterative Algorithms

In the previous section we introduced an algorithm for "exactly" computing the maximum likelihood estimates of the AR coefficients. The primary disadvantage of this algorithm is that one must accurately find the roots of a polynomial whose degree increases exponentially with respect to system order. In this section, we summarize some iterative algorithms that, theoretically, can be used to solve the ML normal equations for AR systems of arbitrary size. These algorithms are used to iteratively solve the exact ML normal equations. Thus, they differ from the recursive algorithms of Burg and Kay, which iteratively maximize an approximation of the true likelihood.

A. Iterative Algorithm for Coupled Systems

In this section, we describe an iterative algorithm that can be extended inductively to systems of arbitrary order. The idea behind the iterative procedure is to decompose a pth-order estimation problem into a $(p-1)$th-order problem coupled with a first-order problem. In the following we describe this procedure for a second-order AR process and then briefly discuss an inductive extension of the algorithm to higher-order processes.

Recall from Section III-B that the normal equations for the AR coefficients of a second-order process can be written as the following polynomials in a_1

$$p_1(x)|_{x=a_1} = x^3\alpha_3(a_2) + x^2\alpha_2(a_2) + x\alpha_1(a_2) + \alpha_0(a_2)|_{x=a_1} = 0 \tag{33}$$

$$p_2(x)|_{x=a_1} = x^2\beta_2(a_2) + x\beta_1(a_2) + \beta_0(a_2)|_{x=a_1} = 0. \tag{34}$$

These normal equations can also be written as polynomials in a_2:

$$\tilde{p}_1(x)|_{x=a_2} = x^3 c_3(a_1) + x^2 c_2(a_1) + x c_1(a_1) + c_0(a_1)|_{x=a_2} = 0 \tag{35}$$

$$\tilde{p}_2(x)|_{x=a_2} = x^3 d_3(a_1) + x^2 d_2(a_1) + x d_1(a_1) + d_0(a_1)|_{x=a_2} = 0. \tag{36}$$

In the following, we describe an iterative procedure for solving these ML normal equations. For a second-order AR process, the "exact" algorithm described in Section III-B is easily implemented and this iterative algorithm is not really necessary. However, this iterative algorithm can be used as an alternative to the exact procedure. The main intent of this section is to provide intuition about the general iterative procedure we will present in the later stages of this section.

The premise of the iterative algorithm is that, for *fixed* a_2, (33) can be easily solved for the AR coefficient a_1. Similarly, for *fixed* a_1, (36) can be easily solved for the AR coefficient a_2. The algorithm alternately fixes a_1 or a_2 and then obtains a new value for the "free" variable by solving only one normal equation. In essence, this procedure decomposes a second-order AR problem into two coupled first-order problems.

Fig. 5 illustrates this procedure. For fixed a_2, the first normal (33) can be used to generate three potential solutions for the AR coefficient a_1. We discard any potential solutions that are not real-valued and/or that do not generate a minimum phase AR polynomial $A(z) = 1 + a_1 z^{-1} + a_2 z^{-2}$. Typically, only one value will satisfy these criteria. If we vary a_2, this procedure can be used to generate a locus of points in R^2 that are potential maximum likelihood solutions. Denote this locus of points by $p1$. Similarly, we can vary a_1 and use the second normal (36) to generate a different locus of potential ML solutions for a_2. Denote this locus of potential solutions by $p2$. In Fig. 5, we plot these loci for a typical time series data set. The time series was constructed using the AR coefficients $a_1 = -2(.95)\cos(\pi/4) \approx -1.34$ and $a_2 = (.95)^2 \approx 0.9$. The input noise variance was set to $\sigma^2 = 1$ and $N = 5$ data points were used to form $\hat{\mathbf{R}}_p(\mathbf{Y})$. The $p1$ and $p2$ loci must intersect at the maximum likelihood solution. Note that it is not necessary to explicitly compute these loci. They are included in the figure for illustrative purposes only. The curve labeled "Itr. locus" is the trajectory of the AR coefficients generated by iterating between the first and second normal equations. In this example, the initial values of the AR coefficients were obtained from the correlation method of linear prediction. Note that the algorithm converges rapidly for this data set. This convergence characteristic is typical for this algorithm even when the data record is small. Also note that if the initialization values are not "sufficiently" close to the ML solution, then the algorithm may not converge or it may converge to a nonminimum phase solution.

It is simple to inductively extend this algorithm to systems of higher-order. Consider this procedure for a third-order system. Let the normal equation polynomials $p_1(a_1, a_2; a_3)$ and $p_2(a_1, a_2; a_3)$ comprise system S2 coupled with polynomial $p_3(a3; a1, a2)$ as system S1. The iterative algorithm works as follows. Obtain an initial estimate, $\hat{a}_3$, using for example linear prediction. Use this estimate and system S2 (which is now effectively second-order) to obtain estimates $\hat{a}_1$ and $\hat{a}_2$. Note that the algorithm for the system S2 must be slightly modified from the algorithm for a true second-order system. That is, we no longer require $|\hat{a}_2| < 1$, as this is no longer a sufficient condition for stability. Instead we require that $\hat{a}_1$ and $\hat{a}_2$ are such that the third-order AR polynomial $\hat{A}(z) = 1 + \hat{a}_1 z^{-1} + \hat{a}_2 z^{-2} + \hat{a}_3 z^{-3}$ be stable. The iteration

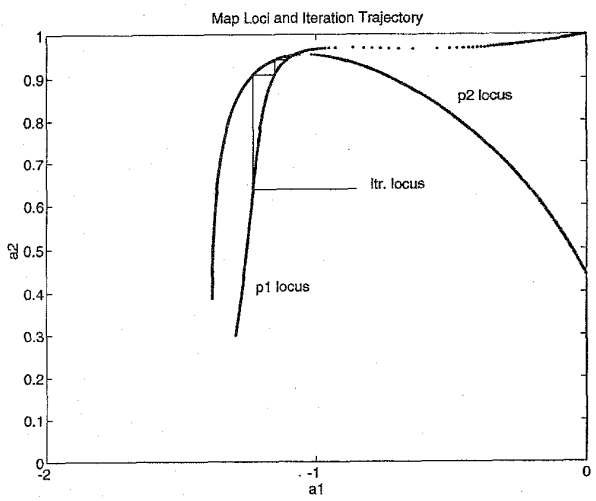


Fig. 5. Typical map loci and iteration trajectory.

is continued by using $\hat{a}_1$ and $\hat{a}_2$ and system S1 to obtain a new estimate for a_3. This procedure is continued until the estimates of the coefficients converge.

B. Newton–Raphson Maps

Let $\mathbf{a}_i = [1\ \theta_i^T]^T$ be the value of the AR coefficients at the ith iteration. Define the error vector at the ith iteration to be

$$\epsilon_i = \mathbf{K}_i \hat{\mathbf{R}}_p(\mathbf{Y})\mathbf{a}_i;$$

where

$$\mathbf{K}_i = [\mathbf{0}\ \mathbf{I}]\left[\mathbf{F}_p\mathbf{F}_p^T - \mathbf{G}_p\mathbf{G}_p^T - \mathbf{a}\mathbf{a}^T + \frac{1}{N}\mathbf{a}_d\mathbf{a}^T\right]\Big|_{\mathbf{a}=\mathbf{a}_i}.$$

The Newton–Raphson map is then given by

$$\theta_{i+1} = \theta_i - \mathbf{H}_i^{-1}\epsilon_i \tag{37}$$

where the matrix $\mathbf{H} = [\mathbf{h}_1\ \cdots\ \mathbf{h}_p]$ has columns

$$\mathbf{h}_k = \frac{\partial \mathbf{K}_i}{\partial a_k}\hat{\mathbf{R}}_p(\mathbf{Y})\mathbf{a}_i + \mathbf{K}_i\hat{\mathbf{R}}_p(\mathbf{Y})\mathbf{e}_{k+1}.$$

Here, $\mathbf{e}_k$ is the kth column of the $(p+1)\times(p+1)$ identity matrix. It is also apparent that

$$\frac{\partial \mathbf{K}_i}{\partial a_k} = [\mathbf{0}\ \mathbf{I}](\mathbf{Z}_k\mathbf{F}_p^T + \mathbf{F}_p\mathbf{Z}_k^T - \mathbf{Z}_{N-k}\mathbf{G}_p^T - \mathbf{G}_p\mathbf{Z}_{N-k}^T + (\mathbf{D}/N - \mathbf{I})(\mathbf{e}_{k+1}\mathbf{a}^T + \mathbf{a}\mathbf{e}_{k+1}^T))$$

where all quantities are evaluated at $\mathbf{a} = \mathbf{a}_i$. In practice, the map of (37) is iterated with a scaled adjustment term $q\mathbf{H}_i^{-1}\epsilon_i$. Each iteration begins with $q = 1$, or with a full correction. If the new value $\mathbf{a}_{i+1}$ generates an error vector ϵ_{i+1} with larger norm than the previous error ϵ_i, then q is reduced by a factor of two and the iteration is performed again.

V. Conclusion

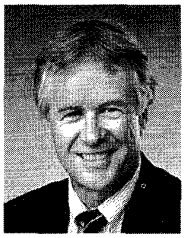
In this paper, we have presented a new algorithm for obtaining the ML estimators of autoregressive time series parameters. This was accomplished by deriving an original set of nonlinear normal equations in the AR coefficients. These normal equations illustrate both the points of contact and divergence between the theory of least squares and the theory of maximum likelihood for time series problems. We have also described an algorithm that solves the nonlinear normal equations for low-order systems. The algorithm consists of finding the roots of a series of polynomials, and choosing the appropriate AR coefficients from this finite set of roots. We have also described some iterative procedures that can be used to solve the ML normal equations.

References

[1] H. Akaike, "Maximum likelihood identification of Gaussian autoregressive moving average models," *Biometrika*, vol. 60, pp. 255–265, 1973.
[2] G. F. Ansley, "An algorithm for the exact likelihood of a mixed autoregressive moving average process," *Biometrika*, vol. 66, pp. 59–65, 1979.
[3] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*. San Francisco, CA: Holden-Day.
[4] J. P. Burg, "Maximum entropy spectrum analysis," Ph.D. dissertation, Stanford Univ., Stanford, CA, 1975.
[5] J. P. Burg, D. G. Luenburger, and D. L. Wenger, "Estimation of structured covariance matrices," *Proc. IEEE*, vol. 70, pp. 963–974, Sept. 1982.
[6] C. J. Demeure and L. L. Scharf, "Sliding windows and lattice algorithms for computing QR factors in the least squares theory of linear prediction," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, pp. 721–725, Apr. 1990.
[7] J. P. Dugre, L. L. Scharf, and C. Gueguen, "Exact likelihood for stationary autoregressive moving average processes," *Signal Processing*, vol. 11, pp 105–118, 1986.
[8] J. Durbin, "The fitting of time series models," *Rev. Intern. Stat. Inst.*, vol. 28, pp. 233–244, 1960.
[9] B. Friedlander, B. M. Morf, T. Kailath, and L. Ljung, "New inversion formulas for matrices classified in terms of their distance from Toeplitz matrices," *Linear Algebra Applications*, vol. 27, pp 31–60, 1979.
[10] G. Heinig and K. Rost, *Algebraic Methods for Toeplitz-Like Matrices and Operators*. Berlin: Birkhauser, 1984.
[11] T. Kailath, B. Levy, L. Ljung, and M. Morf, "Fast time invariant implementation of Gaussian signal detectors," *IEEE Trans. Inform. Theory*, vol. IT-24, July 1978.
[12] S. M. Kay, "Recursive maximum likelihood estimation of autoregressive processes," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 56–65, Feb. 1983.
[13] J. LeRoux and C. J. Gueguen, "A fixed point computation of partial correlation coefficients," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-25, pp. 257–259, 1977.
[14] L. T. McWhorter, "Representations for covariance bounds and time series estimators," Ph. D. dissertation, Univ. of Colorado, Boulder, CO, 1994.
[15] M. Morf, B. Dickinson, T. Kailath, and A. Viera, "Efficient solution of covariance equations for linear prediction," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 25, pp. 429–433, Oct. 1977.
[16] M. Morf, G. S. Sidhu, and T. Kailath, "Some new algorithms for recursive estimation on constant, linear discrete time systems," *IEEE Trans. Automat. Contr.*, vol. AC-19, pp 315–323, Aug. 1974.
[17] C. T. Mullis, Course notes ECEN 5442, Univ. of Colorado, Fall, 1992.
[18] F. C. Schweppe, "Evaluation of likelihood functions for Gaussian signals," *IEEE Trans. Inform. Theory*, vol. IT-11, pp. 61–70, 1965.
[19] M. L. Vis and L. L. Scharf, "A note on recursive maximum likelihood for autoregressive modeling," *IEEE Trans. Signal Processing*, vol. 42, Oct. 1994.
[20] G. Walker, "On periodicity in series of related terms," *Proc. Royal Soc.*, 1931, vol. 131, p. 518.
[21] G. U. Yule, "On a method of investigating periodicities in disturbed series, with special reference to Wolfer's sunspot number," *Phil. Trans.*, vol. 226, p. 267, 1927.

L. Todd McWhorter received the B.S. degree in electrical engineering from Colorado State University, Fort Collins, USA, in 1984. He received the M.S. and Ph.D. degrees in electrical engineering from the University of Colorado in 1991 and 1994, respectively.

Currently, he is employed as a research associate and instructor at the University of Colorado. From 1985 to 1989, he was employed by Woodward Governor Company, Fort Collins, CO, USA, as an applications engineer and circuit designer. His current research interests include time series and array processing algorithms, multiwindow estimators of second- and higher order statistics, covariance bound theory, and spectral estimation.



Louis L. Scharf (F'86) received the Ph.D. degree in electrical engineering from the University of Washington, Seattle, USA, in 1969.

From 1969 to 1971, he was a Member of the Technical Staff at Honeywell's Marine Systems Center in Seattle. He was a Faculty Member at Colorado State University, Fort Collins, USA, from 1971 to 1981, where he last served as a Professor of Electrical Engineering and Statistics. From 1982 to 1985, he was a Professor and Chair of Electrical Engineering at the University of Rhode Island, Kingston, USA. He is currently a Professor of Electrical and Computer Engineering at the University of Colorado, Boulder, USA, where he teaches and conducts research in signal processing. In 1974, he was a Visiting Associate Professor at Duke University, Durham, NC, USA. In 1977, he was at the University of South Paris, Orsay, France, where he was a Member of the Technical Staff in the CNRS Laboratoire des Signaux et Systemes, Gif-sur-Yvette. In 1981, he was a Visiting Professor at the University of La Plata, Argentina, and a Visiting Professor at the Ecole Nationale Supérieure des Télécommunications, Paris, France. He has served as a Consultant to Honeywell, Inc., the Applied Physics Labs Seattle, the Research Triangle Institute, Green Mountain Geophysics, and Ball Aerospace Corporation.

Dr. Scharf is a past member of the Acoustics, Speech, and Signal Processing ADCOM and the Editorial Board of *Signal Processing*. He is a past Associate Editor for the IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING. He was the Technical Program Chairman for the 1980 International Conference on Acoustics, Speech, and Signal Processing. He was appointed an IEEE National Distinguished Lecturer for 1993–1994 and is a member of Eta Kappa Nu.