AN ALGORITHM FOR THE EXACT LIKELIHOOD OF A STATIONARY VECTOR AUTOREGRESSIVE-MOVING AVERAGE MODEL

By José Alberto Mauricio

Universidad Complutense de Madrid

First Version received June 1999

Abstract. The so-called innovations form of the likelihood function implied by a stationary vector autoregressive-moving average model is considered without directly using a state–space representation. Specifically, it is shown in detail how to compute the exact likelihood by an adaptation to the multivariate case of the innovations algorithm of Ansley (1979) for univariate models. Comparisons with other existing methods are also provided, showing that the algorithm described here is computationally more efficient than the fastest methods currently available in many cases of practical interest.

Keywords. Innovations; exact likelihood function; exact maximum likelihood estimation; time series; vector autoregressive-moving average model.

1. INTRODUCTION

Let a multivariate (vector) time series $\mathbf{w}_1, \mathbf{w}_2 \ldots, \mathbf{w}_n$ be generated by an m-dimensional ($m \geqslant 2$), stationary, Gaussian process $\{\mathbf{W}_t\}_{t=0,\pm1,\ldots}$, assumed to be adequately represented by a vector autoregressive-moving average (ARMA) model of the form

$$\mathbf{\Phi}(B)\tilde{\mathbf{W}}_t = \mathbf{\Theta}(B)\mathbf{A}_t \tag{1}$$

where

$$\mathbf{\Phi}(B) = \mathbf{I} - \mathbf{\Phi}_1 B - \cdots - \mathbf{\Phi}_p B^p$$

$$\mathbf{\Theta}(B) = \mathbf{I} - \mathbf{\Theta}_1 B - \cdots - \mathbf{\Theta}_q B^q$$

B is the usual backshift operator (i.e., $B^k\tilde{\mathbf{W}}_t = \tilde{\mathbf{W}}_{t-k}$); $\mathbf{\Phi}_i (i = 1, \ldots, p)$, $\mathbf{\Theta}_i (i = 1, \ldots, q)$ and $\boldsymbol{\mu}$ are $m \times m$, $m \times m$ and $m \times 1$ parameter matrices, respectively; $\tilde{\mathbf{W}}_t = \mathbf{W}_t - \boldsymbol{\mu}$, with $\boldsymbol{\mu} = E[\mathbf{W}_t]$; and $\{\mathbf{A}_t\}_{t=0,\pm1,\ldots}$ is an m-dimensional zero-mean Gaussian white-noise process with covariance matrix $\sigma^2\mathbf{Q}$, where $\sigma^2 > 0$ and $\mathbf{Q}(m \times m)$ is symmetric positive definite. For stationarity, it is required that the roots of $|\mathbf{\Phi}(B)| = 0$ lie outside the unit circle. In what follows, all of the elements of $\mathbf{\Phi}_i (i = 1, \ldots, p)$, $\mathbf{\Theta}_i (i = 1, \ldots, q)$, $\boldsymbol{\mu}$ and $\mathbf{Q}$ are stacked together into a single vector $\boldsymbol{\beta}$ for notational convenience.

It may be noted that the specification of $E[\mathbf{A}_t\mathbf{A}_t']$ above as the product $\sigma^2\mathbf{Q}$, is just a useful device for obtaining exact maximum likelihood estimates by

maximizing a concentrated log-likelihood as a function of the parameter vector $\boldsymbol{\beta}$ only. A few comments on this subject are given in Section 3; in particular, the fact that the decomposition $\sigma^2\mathbf{Q}$ is not unique raises no problem in the estimation of $E[\mathbf{A}_t\mathbf{A}_t']$, since, on convergence of the estimation algorithm, interest lies in the product $\sigma^2\mathbf{Q}$, but not in σ^2 nor in $\mathbf{Q}$ by itself; for the purpose of just computing the exact likelihood function, any decomposition of $E[\mathbf{A}_t\mathbf{A}_t']$ is valid.

As noted above, given an observed time series $\mathbf{w} = (\mathbf{w}_1', \ldots, \mathbf{w}_n')'$ consisting of n $m \times 1$ data vectors, the functional form of the corresponding likelihood function $L(\boldsymbol{\beta}, \sigma^2|\mathbf{w})$ is required for computing exact maximum likelihood estimates of $\boldsymbol{\beta}$ and σ^2. The likelihood function has exactly the same form as the density function of the $mn \times 1$ random vector $\mathbf{W} = (\mathbf{W}_1', \ldots, \mathbf{W}_n')'$; hence, for a Gaussian process, the 'direct' form of $L(\boldsymbol{\beta}, \sigma^2|\mathbf{w})$ is

$$L(\boldsymbol{\beta}, \sigma^2|\mathbf{w}) = (2\pi\sigma^2)^{-mn/2}|\boldsymbol{\Sigma}_{\mathbf{W}}|^{-1/2}\exp\left\{-(2\sigma^2)^{-1}\tilde{\mathbf{w}}'\boldsymbol{\Sigma}_{\mathbf{W}}^{-1}\tilde{\mathbf{w}}\right\} \tag{2}$$

where $\tilde{\mathbf{w}} = (\tilde{\mathbf{w}}_1, \ldots, \tilde{\mathbf{w}}_n')'$, $\tilde{\mathbf{w}}_t = \mathbf{w}_t - \boldsymbol{\mu}(t = 1, \ldots, n)$, $\boldsymbol{\Sigma}_{\mathbf{W}} = \sigma^{-2}E[\tilde{\mathbf{W}}\tilde{\mathbf{W}}']$ (which depends only on $\boldsymbol{\beta}$), and $\tilde{\mathbf{W}} = (\tilde{\mathbf{W}}_1, \ldots, \tilde{\mathbf{W}}_n')'$. For practical purposes, the form (2) of $L(\boldsymbol{\beta}, \sigma^2|\mathbf{w})$ has two main drawbacks:

(i) It explicitly requires computing the theoretical autocovariance matrices of orders 0 through $n - 1$ implied by (1).
(ii) It implicitly requires computing both the inverse and the determinant of a symmetric matrix of order mn.

Many authors have derived explicit forms for $\tilde{\mathbf{w}}'\boldsymbol{\Sigma}_{\mathbf{W}}^{-1}\tilde{\mathbf{w}}$ and $|\boldsymbol{\Sigma}_{\mathbf{W}}|$, in both the scalar ($m = 1$) and vector ($m \geqslant 2$) cases, which require much less computational effort than a direct evaluation.

In the scalar case, methods based on different implementations of the Kalman filter applied to a suitable state–space representation of (1), are often considered the most efficient ones on the basis of what are usually known as 'theoretical operation counts'. In particular, Pearlman (1980, p. 232) states that, to compute the exact likelihood function when $m = 1$

$$N_A = n\left\{p + \frac{(q+1)(q+4)}{2}\right\}$$

time consuming operations (multiplications or divisions involving floating-point numbers) are required by the algorithm of Ansley (1979); note that N_A is quadratic in q (the order of the moving average part of the model) but only linear in p (the order of the autoregressive part of the model). Also, Kohn and Ansley (1985, p. 229) state that the algorithm of Pearlman (1980) as implemented in Mélard (1984), requires

$$N_{PM} = n(2p + 3q + 2)$$

operations when $p < q + 1$, and

$$N'_{PM} = n(p + 4q + 6)$$

when $p \geqslant q + 1$; see also Mélard (1984, p. 108). Finally, except for the last p observations, the method of Kohn and Ansley (1985) requires

$$N_{KA} = n(p + 3q + 2)$$

operations. These theoretical counts have often been used to claim that methods based on the Kalman filter – i.e., Pearlman (1980), Mélard (1984) and Kohn and Ansley (1985) – are the most efficient ones in the scalar case.

However, it should be noted that, in addition to either N_{PM}, N'_{PM} or N_{KA}, every method based on the Kalman filter for scalar models requires a significant number of operations to start the filtering recursions. This number of 'preliminary' operations is not so siginificant for other methods, such as the ones of Ansley (1979) and Ljung and Box (1979). Specifically, to start the filtering recursions, methods based on the Kalman filter require computing the theoretical autocovariances of orders 0 through g, with $g = \max(p, q)$, whereas the methods of Ansley (1979) and Ljung and Box (1979) require computing the theoretical autocovariances of orders 0 through only $p - 1$; in particular, no theoretical autocovariances are required by methods outside the state–space framework when $p = 0$. This fact implies that:

(i) with regard to the number of preliminary operations, computational savings can be acheived outside the state–space framework when $q > p$ (especially if q is large and much greater that p) that mitigate, to some extent, the effects of N_A being quadratic in q, and
(ii) a comparison based solely on theoretical counts such as N_A, N_{PM}, N'_{PM} and N_{KA}, can be misleading, since those counts ignore the computational burden required, for instance, to start the Kalman filter.

As an example of this situation, Mélard (1984, p. 108) states that such burden is quadratic in p, q and g, although this fact is ignored in theoretical comparisons with other methods presented in that paper. Furthermore, one of the main reasons of the computational efficiency of the methods of Mélard (1984) and Kohn and Ansley (1985) for scalar models, is that both methods use a special algorithm for computing the theoretical autocovariance function, which cannot be adapted for use in the multivariate case; see, for example, Tunnicliffe-Wilson (1979), and note that Mélard (1984, pp. 108–9) compares his algorithm with the method of Ansley (1979) 'improved' with the algorithm of McLeod (1975), which is not as efficient as that of Tunnicliffe-Wilson (1979).

The discussion above implies that:

(i) there may exist practical situations in which the method of Ansley (1979) can be implemented to perform more efficiently than methods based on the Kalman filter, even when q is moderately large, in both the scalar and vector cases, and

(ii) comparisons based solely on theoretical counts can be misleading and shouls be avoided in favour of those based on actual counts.

Furthermore, actual counts are much easier to compute accurately than theoretical counts, since actual counts can be obtained simply by using an integer counter within the actual program implementing whatever algorithm, which is initialized to zero and increases by one unit whenever either a multiplication or a division involving floating-point numbers is performed. Actual operation counts for several values of $m \geqslant 2, n, p$ and q, have been used in Mauricio (1995, 1997) to study the relative computational efficiency of different algorithms for computing the exact likelihood implied by vector ARMA models. Results reported in those papers show that the algorithms of Shea (1989) and Mauricio (1997) are the most efficient ones previously given in the literature.

This paper presents a detailed alternative to currently existing methods for computing both $\tilde{\mathbf{w}}'\boldsymbol{\Sigma}_{\mathbf{W}}^{-1}\tilde{\mathbf{w}}$ and $|\boldsymbol{\Sigma}_{\mathbf{W}}|$ in (2) – one which does not seem to have been published previously. The method presented here is an adaptation of the algorithm due to Ansley (1979) for scalar models which, in turn, is an example of a class of methods which are based on the so-called 'innovations' form of $L(\boldsymbol{\beta}, \sigma^2|\mathbf{w})$. The innovations form of the exact likelihood for ARMA models is discussed, for example, in Box, et al. (1994, ch. 7) and in Reinsel (1997, ch. 5). In particular, Reinsel (1997, pp. 145–7) describes a method for computing the exact likelihood in the multivariate case, which starts being similar to that of Ansley (1979), but then proceeds, in a rather complicated way, being similar to recursive methods based on the Kalman filter; note also that details on the practical implementation of the computations required by the method of Reinsel (1997) are omitted in that book.

From a notational standpoint, a very simple innovations form of the exact likelihood function of vector ARMA models is derived in Section 2 of this paper. In Section 3, detailed descriptions are given for computing the elements of the form of $L(\boldsymbol{\beta}, \sigma^2|\mathbf{w})$ derived in Section 2. A comparison between the algorithm presented in Section 3 and other existing methods for computing $L(\boldsymbol{\beta}, \sigma^2|\mathbf{w})$ is given in Section 4. Finally, in Section 5, conclusions are summarized.

2. AN INNOVATIONS FORM OF THE EXACT LIKELIHOOD FUNCTION

Consider a given time series $\mathbf{w} = (\mathbf{w}_1', \ldots, \mathbf{w}_n')'$ and let $\tilde{\mathbf{W}} = (\tilde{\mathbf{W}}_1', \ldots, \tilde{\mathbf{W}}_n')'$ (an $mn \times 1$ vector), $\mathbf{A} = (\mathbf{A}_1', \ldots, \mathbf{A}_n')'$ (an $mn \times 1$ vector) and $\mathbf{U}_* = (\tilde{\mathbf{W}}_{1-p,\ldots,}'\tilde{\mathbf{W}}_0', \mathbf{A}_{1-q,\ldots,}'\mathbf{A}_0')'$ (an $m(p+q) \times 1$ vector). Then, using (1) with $t = 1, \ldots, n$, $\mathbf{w}$ can be regarded as a particular realization of a random vector $\mathbf{W}$ which can be adequately represented by the model

$$\mathbf{D}_{\boldsymbol{\Phi}}\tilde{\mathbf{W}} = \mathbf{D}_{\boldsymbol{\Theta}}\mathbf{A} + \mathbf{V}\mathbf{U}_* \tag{3}$$

where $\mathbf{D}_{\mathbf{\Phi}}$ and $\mathbf{D}_{\mathbf{\Theta}}$ are $mn \times mn$ lower triangular block-matrices with identity matrices of order m on the main diagonal, and $-\mathbf{\Phi}_k$ and $-\mathbf{\Theta}_k$, respectively, down the kth subdiagonal, and $\mathbf{V}$ is an $mn \times m(p+q)$ block-matrix which can be partitioned as $\mathbf{V} = [\mathbf{G}_{\mathbf{\Phi}}\ \mathbf{G}_{\mathbf{\Theta}}]$, with

$$\underset{mn\times mp}{\mathbf{G}_{\mathbf{\Phi}}} = \begin{bmatrix} \mathbf{\Phi}_p & \mathbf{\Phi}_{p-1} & \mathbf{\Phi}_{p-2} & \cdots & \mathbf{\Phi}_1 \\ \mathbf{0} & \mathbf{\Phi}_p & \mathbf{\Phi}_{p-1} & \cdots & \mathbf{\Phi}_2 \\ \mathbf{0} & \mathbf{0} & \mathbf{\Phi}_p & \cdots & \mathbf{\Phi}_3 \\ \vdots & \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{\Phi}_p \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}$$

$$\underset{mn\times mq}{\mathbf{G}_{\mathbf{\Theta}}} = \begin{bmatrix} -\mathbf{\Theta}_q & -\mathbf{\Theta}_{q-1} & -\mathbf{\Theta}_{q-2} & \cdots & -\mathbf{\Theta}_1 \\ \mathbf{0} & -\mathbf{\Theta}_q & -\mathbf{\Theta}_{q-1} & \cdots & -\mathbf{\Theta}_2 \\ \mathbf{0} & \mathbf{0} & -\mathbf{\Theta}_q & \cdots & -\mathbf{\Theta}_3 \\ \vdots & \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & -\mathbf{\Theta}_q \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}$$

Note that letting $g = \max(p, q)$, $\mathbf{V}$ can also partitioned as $\mathbf{V} = [\mathbf{V}_1'\ \mathbf{0}]'$, with the $mg \times m(p+q)$ matrix $\mathbf{V}_1$ containing the first mg rows of $\mathbf{V}$. If an $m(p+q) \times m(p+q)$ matrix $\mathbf{\Omega}$ is now defined by $E[\mathbf{U}_*\mathbf{U}_*'] = \sigma^2\mathbf{\Omega}$, then, noting that $\mathbf{A} \sim \mathbf{N}[\mathbf{0}, \sigma^2(\mathbf{I}_n \otimes \mathbf{Q})], \mathbf{U}_* \sim \mathbf{N}(\mathbf{0}, \sigma^2\mathbf{\Omega})$ and $E[\mathbf{U}_*\mathbf{A}'] = \mathbf{0}$, it follows from (3) that the random vector $\mathbf{Y} = \mathbf{D}_{\mathbf{\Phi}}\tilde{\mathbf{W}}$ has a zero-mean normal distribution with covariance matrix $E[\mathbf{Y}\mathbf{Y}'] = \sigma^2\mathbf{\Sigma}_{\mathbf{Y}}$, where

$$\begin{aligned} \mathbf{\Sigma}_{\mathbf{Y}} &= \mathbf{D}_{\mathbf{\Theta}}(\mathbf{I}_n \otimes \mathbf{Q})\mathbf{D}_{\mathbf{\Theta}}' + \mathbf{V}\mathbf{\Omega}\mathbf{V}' \\ &= \mathbf{D}_{\mathbf{\Theta}}(\mathbf{I}_n \otimes \mathbf{Q})\mathbf{D}_{\mathbf{\Theta}}' + \begin{bmatrix} \mathbf{V}_1\mathbf{\Omega}\mathbf{V}_1' & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \end{aligned}$$

Recalling the special structure of $\mathbf{D}_{\mathbf{\Theta}}$, the only nonzero $m \times m$ blocks that form the main diagonal and the lower triangle of the symmetric matrix $\mathbf{D}_{\mathbf{\Theta}}(\mathbf{I}_n \otimes \mathbf{Q})\mathbf{D}_{\mathbf{\Theta}}'$ are given by

$$[\mathbf{D}_{\mathbf{\Theta}}(\mathbf{I}_n \otimes \mathbf{Q})\mathbf{D}_{\mathbf{\Theta}}']_{ij} = \begin{cases} \displaystyle\sum_{k=0}^{j-1} \mathbf{\Theta}_{k+i-j}\mathbf{Q}\mathbf{\Theta}_k' & \text{for } i = 1, \ldots, q+1; j = 1, \ldots, i \\ [\mathbf{D}_{\mathbf{\Theta}}(\mathbf{I}_n \otimes \mathbf{Q})\mathbf{D}_{\mathbf{\Theta}}']_{i-1,j-1} & \text{for } i = q+2, \ldots, n; j = i-q, \ldots, i \end{cases} \tag{5}$$

where the convention $\mathbf{\Theta}_0 = -\mathbf{I}_m$ is used. Hence, considering $\mathbf{\Sigma_Y}$ as an $n \times n$ array whose elements are $m \times m$ blocks, it follows from (4) and (5) that $\mathbf{\Sigma_Y}$ is a block-band matrix with maximum block-bandwidth g for the first g block-rows, and block-bandwidth q thereafter. A similar proof of this result can be found in Reinsel (1997, p. 145).

As an illustration, consider an m-variate ARMA(3,1) model. In this case, it can be seen, from (4) and (5) with $p = 3$ and $q = 1$, that

$$\Sigma_{\mathbf{Y}} = \begin{bmatrix} \mathbf{X}_{11} & \mathbf{X}_{21}' & \mathbf{V}_{31}' & & & & & & \\ \mathbf{X}_{21} & \mathbf{X}_{22} & \mathbf{X}_{32}' & & & & & & \\ \mathbf{V}_{31} & \mathbf{X}_{32} & \mathbf{X}_{33} & \mathbf{D}_{43}' & & & & & \\ & & \mathbf{D}_{43} & \mathbf{D}_{44} & \mathbf{D}_{43}' & & & & \\ & & & \ddots & \ddots & \ddots & & & \\ & & & & \mathbf{D}_{43} & \mathbf{D}_{44} & \mathbf{D}_{43}' & & \\ & & & & & \mathbf{D}_{43} & \mathbf{D}_{44} & \mathbf{D}_{43}' & \\ & & & & & & \mathbf{D}_{43} & \mathbf{D}_{44} \end{bmatrix} \tag{6}$$

where $\mathbf{V}_{ij}$ represents an $m \times m$ block whose elements depend on $\mathbf{V}_1 \mathbf{\Omega} \mathbf{V}_1'$ only, $\mathbf{D}_{ij}$ represents an $m \times m$ block whose elements depend on $\mathbf{D}_{\mathbf{\Theta}}(\mathbf{I}_n \otimes \mathbf{Q})\mathbf{D}_{\mathbf{\Theta}}'$ only, and $\mathbf{X}_{ij}$ represents an $m \times m$ block whose elements depend on both of $\mathbf{V}_1 \mathbf{\Omega} \mathbf{V}_1'$ and $\mathbf{D}_{\mathbf{\Theta}}(\mathbf{I}_n \otimes \mathbf{Q})\mathbf{D}_{\mathbf{\Theta}}'$. As a further example, consider an m-variate ARMA(2,2) model. In this case, it can be seen, from (4) and (5) with $p = q = 2$, that

$$\Sigma_{\mathbf{Y}} = \begin{bmatrix} \mathbf{X}_{11} & \mathbf{X}_{21}' & \mathbf{D}_{31}' & & & & & \\ \mathbf{X}_{21} & \mathbf{X}_{22} & \mathbf{D}_{32}' & \mathbf{D}_{31}' & & & & \\ \mathbf{D}_{31} & \mathbf{D}_{32} & \mathbf{D}_{33} & \mathbf{D}_{32}' & \mathbf{D}_{31}' & & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & & \\ & & \mathbf{D}_{31} & \mathbf{D}_{32} & \mathbf{D}_{33} & \mathbf{D}_{32}' & \mathbf{D}_{31}' \\ & & & \mathbf{D}_{31} & \mathbf{D}_{32} & \mathbf{D}_{33} & \mathbf{D}_{32}' \\ & & & & \mathbf{D}_{31} & \mathbf{D}_{32} & \mathbf{D}_{33} \end{bmatrix} \tag{7}$$

The illustrative forms of $\Sigma_{\mathbf{Y}}$ in (6) and (7) are useful for understanding the need for a special procedure for factoring $\mathbf{\Sigma_Y}$, which is discussed in Section 3.

Returning to the random vector $\mathbf{Y} = \mathbf{D}_{\mathbf{\Phi}}\tilde{\mathbf{W}}$ and recalling that $\mathbf{D}_{\mathbf{\Phi}}$ is a unit lower triangular matrix, it turns out that the Jacobian of the transformation $\tilde{\mathbf{W}} = \mathbf{D}_{\mathbf{\Phi}}^{-1}\mathbf{Y}$ is unity, so that the density function for $\tilde{\mathbf{W}}$ has the same form as the density function for $\mathbf{Y}$. Thus, the exact likelihood function (2) can be written as

$$L(\boldsymbol{\beta}, \sigma^2 | \mathbf{w}) = (2\pi\sigma^2)^{-mn/2} |\mathbf{\Sigma_Y}|^{-1/2} \exp\{-(2\sigma^2)^{-1}\mathbf{y}'\mathbf{\Sigma_Y}^{-1}\mathbf{y}\} \tag{8}$$

where $\mathbf{y} = \mathbf{D}_{\mathbf{\Phi}}\tilde{\mathbf{w}}$ is a vector of 'derived' observations, in the sense that $\mathbf{y}$ is derived from the original time series $\mathbf{w}$ through a simple linear transformation. From a computational standpoint, the form (8) of the required likelihood function has two important advantages over the direct form (2):

(i) It explicitly requires computing the theoretical autocovariance matrices implied by (1) of orders 0 through only $p-1$.
(ii) It implicitly requires computing both the inverse and the determinant of a symmetric block-band matrix, which has the vast majority of its elements equal to zero.

If $\mathbf{L}$ represents the Cholesky factor of $\mathbf{\Sigma_Y}$ (i.e., $\mathbf{\Sigma_Y} = \mathbf{LL}'$ with $\mathbf{L}$ lower triangular, having exactly the same block-band structure as $\mathbf{\Sigma_Y}$), then (8) can be finally written as

$$L(\boldsymbol{\beta}, \sigma^2|\mathbf{w}) = (2\pi\sigma^2)^{-mn/2}|\mathbf{L}|^{-1}\exp\{-(2\sigma^2)^{-1}\mathbf{e}'\mathbf{e}\} \tag{9}$$

where $\mathbf{e} = \mathbf{L}^{-1}\mathbf{y} = \mathbf{L}^{-1}\mathbf{D_\Phi}\tilde{\mathbf{w}}$ can be regarded as a particular realization of a random vector $\mathbf{E} = \mathbf{L}^{-1}\mathbf{Y}$ following a $\mathbf{N}(\mathbf{0}, \sigma^2\mathbf{I})$ distribution. Equation (9) is the innovations form of the exact likelihood function which, for $m = 1$, corresponds to equation (3.2) in Ansley (1979).

In the case of pure autoregressive models, it follows from (4) with $q = 0$ (which implies that $\mathbf{D_\Theta} = \mathbf{I}$) that

$$\mathbf{\Sigma_Y} = \begin{bmatrix} (\mathbf{I}_p \otimes \mathbf{Q}) + \mathbf{V}_1\mathbf{\Omega}\mathbf{V}_1' & \mathbf{0} \\ \mathbf{0} & (\mathbf{I}_{n-p} \otimes \mathbf{Q}) \end{bmatrix} \tag{10}$$

which is a block-diagonal matrix. Hence, when $q = 0$ the Cholesky factor of $\mathbf{\Sigma_Y}$ has the form

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_p & \mathbf{0} \\ \mathbf{0} & (\mathbf{I}_{n-p} \otimes \mathbf{R}) \end{bmatrix} \tag{11}$$

where $\mathbf{L}_p$ and $\mathbf{R}$ represent the Cholesky factors of $(\mathbf{I}_p \otimes \mathbf{Q}) + \mathbf{V}_1\mathbf{\Omega}\mathbf{V}_1'$ and $\mathbf{Q}$, respectively. It follows from (4) and (10)–(11) that obtaining $\mathbf{L}$ when $q = 0$ is a much simpler computational task than obtaining $\mathbf{L}$ when $q \neq 0$. Also, note that for $m = 1$, (10) and (11) correspond to equation (3.3) in Ansley (1979). As a final remark, it may be noted that the transformation $\mathbf{Y} = \mathbf{D_\Phi}\tilde{\mathbf{W}}$ used in this paper is not identical to the one used by Ansley (1979), although it is computationally equivalent in the sense that both transformations generate random vectors whose covariance matrices have exactly the same band structure.

3. COMPUTATION OF THE EXACT LIKELIHOOD FUNCTION

The scale parameter σ^2 may be differentiated out of (2) to yield the following concentrated log-likelihood:

$$l_*(\boldsymbol{\beta}|\mathbf{w}) = -\frac{mn}{2}\left[\log(2\pi) - \log(mn) + 1 + \log\{(\tilde{\mathbf{w}}'\mathbf{\Sigma}_\mathbf{W}^{-1}\tilde{\mathbf{w}})|\mathbf{\Sigma_W}|^{1/(mn)}\}\right]$$

Hence, for estimation purposes, maximizing (2) is equivalent to minimizing

$$F(\boldsymbol{\beta}|\mathbf{w}) = (\tilde{\mathbf{w}}'\boldsymbol{\Sigma}_{\mathbf{W}}^{-1}\tilde{\mathbf{w}})|\boldsymbol{\Sigma}_{\mathbf{W}}|^{1/(mn)} \tag{12}$$

where it can be seen from (2), (8) and (9) that

$$\tilde{\mathbf{w}}'\boldsymbol{\Sigma}_{\mathbf{W}}^{-1}\tilde{\mathbf{w}} = \mathbf{y}'\boldsymbol{\Sigma}_{\mathbf{Y}}^{-1}\mathbf{y} = \mathbf{e}'\mathbf{e} = \sum_{t=1}^{mn} e_t^2 \tag{13}$$

and that

$$|\boldsymbol{\Sigma}_{\mathbf{W}}| = |\boldsymbol{\Sigma}_{\mathbf{Y}}| = |\mathbf{L}|^2 = \prod_{t=1}^{mn} L_{tt}^2 \tag{14}$$

It is now shown in detail how to compute $\mathbf{e}$ and $\mathbf{L}$ in (13)–(14) which, in turn, can be used to evaluate numerically (9) and (12). Recall that $\mathbf{e}$ is the solution to $\mathbf{Le} = \mathbf{y}$, with $\mathbf{y} = \mathbf{D}_{\boldsymbol{\Phi}}\tilde{\mathbf{w}}$, and that $\mathbf{L}$ is the Cholesky factor of the symmetric matrix given in (4). To evaluate (4), note that $\boldsymbol{\Omega} = \sigma^{-2}E[\mathbf{U}_*\mathbf{U}_*']$ with $\mathbf{U}_* = (\tilde{\mathbf{W}}_{1-p}', \ldots, \tilde{\mathbf{W}}_0', \mathbf{A}_{1-q}', \ldots, \mathbf{A}_0')'$, so that

$$\boldsymbol{\Omega} = \begin{bmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{C}' & (\mathbf{I}_q \otimes \mathbf{Q}) \end{bmatrix}$$

where the (i,j)th block of $\mathbf{B}$ is

$$\begin{aligned} \mathbf{B}_{ij} &= \sigma^{-2}E\left[\tilde{\mathbf{W}}_{i-p}\tilde{\mathbf{W}}_{j-p}'\right] \\ &= \boldsymbol{\Gamma}_{j-i} \qquad \text{for} \quad i,j = 1,\ldots,p \end{aligned}$$

and the (i,j)th block of $\mathbf{C}$ is

$$\begin{aligned} \mathbf{C}_{ij} &= \sigma^{-2}E\left[\tilde{\mathbf{W}}_{i-p}\mathbf{A}_{j-q}'\right] \\ &= \boldsymbol{\Lambda}_{j-i-q+p} \qquad \text{for } i = 1,\ldots,p;\ j = 1,\ldots,q \end{aligned}$$

Because $\mathbf{B}$ is symmetric and $E\left[\tilde{\mathbf{W}}_{t-i}\mathbf{A}_t'\right] = \mathbf{0}$ for $i > 0$, to compute $\mathbf{B}$ and $\mathbf{C}$ only the theoretical autocovariance matrices $\boldsymbol{\Gamma}_k$ for $k = 0, 1, \ldots, p-1$, and the theoretical cross-covariance matrices $\boldsymbol{\Lambda}_k$ for $k = 0, -1, \ldots, -q+1$, are needed. An efficient method for carrying out those computations can be found in Mauricio (1997). Note that existing methods based on the Kalman filter – e.g., Shea (1989) – require computing $\boldsymbol{\Gamma}_k$ for $k = 0, 1, \ldots, g$ to start the filtering recursions, which implies a computational overload with respect to other methods when q is large and much greater than p (in particular, when $p = 0$).

Once the elements of $\boldsymbol{\Omega}$ are available, the (i,j)th block $(i = 1,\ldots,g; j = 1,\ldots,i)$ of the symmetric matrix $\mathbf{V}_1\boldsymbol{\Omega}\mathbf{V}_1'$ in (4) is given by

$$(\mathbf{V}_1\boldsymbol{\Omega}\mathbf{V}_1')_{ij} = \sum_{k=0}^{p-i} \boldsymbol{\Phi}_{p-k}\mathbf{F}_{k+i,j} - \sum_{k=0}^{q-i} \boldsymbol{\Theta}_{q-k}\mathbf{F}_{k+p+i,j} \tag{15}$$

where, for $j = 1,\ldots,g$,

$$\mathbf{F}_{ij} = \begin{cases} \sum_{k=j-i}^{p-i} \boldsymbol{\Gamma}_k \boldsymbol{\Phi}'_{p-k-i+j} - \sum_{k=j-i}^{q-i} \boldsymbol{\Lambda}_{p-q+k} \boldsymbol{\Theta}'_{q-k-i+j} & i = 1, \ldots, p \\ \sum_{k=p+j-i}^{2p-i} \boldsymbol{\Lambda}'_{p-q-k} \boldsymbol{\Phi}'_{2p-k-i+j} - \mathbf{Q} \boldsymbol{\Theta}'_{q+p-i+j} & i = p+1, \ldots, p+q \end{cases}$$

with $\boldsymbol{\Gamma}_k = \boldsymbol{\Gamma}'_{-k}$ for $k < 0$, $\boldsymbol{\Lambda}_k = \mathbf{0}$ for $k > 0$ and $\boldsymbol{\Theta}_i = \mathbf{0}$ for $i > q$.

The only nonzero $m \times m$ blocks that form the main diagonal and the lower triangle of the symmetric matrix $\mathbf{D}_{\boldsymbol{\Theta}}(\mathbf{I}_n \otimes \mathbf{Q})\mathbf{D}'_{\boldsymbol{\Theta}}$ in (4), can be computed using (5). Alternatively, letting $\mathbf{H}_0 = -\mathbf{Q}$ and $\mathbf{H}_i = \boldsymbol{\Theta}_i \mathbf{Q} (i = 1, \ldots, q)$, it follows from (5) that

$$\left[\mathbf{D}_{\boldsymbol{\Theta}}(\mathbf{I} \otimes \mathbf{Q})\mathbf{D}'_{\boldsymbol{\Theta}}\right]_{ij} = \begin{cases} -\mathbf{H}_{i-j} + \sum_{k=1}^{j-1} \mathbf{H}_{k+i-j} \boldsymbol{\Theta}'_k & \text{for } i = 1, \ldots, q+1;\ j = 1, \ldots, i \\ \left[\mathbf{D}_{\boldsymbol{\Theta}}(\mathbf{I} \otimes \mathbf{Q})\mathbf{D}'_{\boldsymbol{\Theta}}\right]_{i-1, j-1} & \text{for } i = q+2, \ldots, n;\ j = i-q, \ldots, i \end{cases} \tag{16}$$

which may save a few operations in comparison to a direct implementation of (5). Equations (15) and (16) can be used to compute the nonzero elements of the band matrix $\boldsymbol{\Sigma}_{\mathbf{Y}}$ in (4) which, in turn, has to be factored to obtain $\mathbf{L}$ and $|\boldsymbol{\Sigma}_{\mathbf{Y}}| = |\mathbf{L}|^2$. It should be noted at this point that $\boldsymbol{\Sigma}_{\mathbf{Y}}$ is not a band matrix in the usual sense; hence, if one wishes to completely exploit the special structure of $\boldsymbol{\Sigma}_{\mathbf{Y}}$, an algorithm for factoring symmetric band matrices, such as the one published by Martin and Wilkinson (1965), should not be used directly. To see this, recall the examples given in (6) and (7); taking $m = 2$, it can be seen, for instance, that the seventh row of $\boldsymbol{\Sigma}_{\mathbf{Y}}$ in (6) has only two nonzero elements to the left of the corresponding diagonal element, whereas it has three nonzero elements to the right. Also, taking $m = 4$, it can be seen, for instance, that the twelfth row of $\boldsymbol{\Sigma}_{\mathbf{Y}}$ in (7) has eleven nonzero elements to the left of the corresponding diagonal element, whereas it only has eight nonzero elements to the right. Hence, considering $\boldsymbol{\Sigma}_{\mathbf{Y}}$ as an $mn \times mn$ array, it follows that $\boldsymbol{\Sigma}_{\mathbf{Y}}$ is a special band matrix with different left and right bandwidths (a situation that does not occur when $m = 1$) which, furthermore, vary from the first mg rows to the remaining $m(n-g)$ rows (a situation that is shared with the case $m = 1$).

I have devised a suitable algorithm for computing the Cholesky factor $\mathbf{L}$ of $\boldsymbol{\Sigma}_{\mathbf{Y}}$ and $|\boldsymbol{\Sigma}_{\mathbf{Y}}| = |\mathbf{L}|^2$, which takes into account the subtleties mentioned above for any values of m, p and q. Basically, given any values for m, p and q, the algorithm keeps tract of the actual left and right bandwidths for each row of $\boldsymbol{\Sigma}_{\mathbf{Y}}$, so that operations involving zero elements are never performed. Using that algorithm results in noticeable computational savings in comparison to a direct use of the algorithm of Martin and Wilkinson (1965), which requires specifying for every row of $\boldsymbol{\Sigma}_{\mathbf{Y}}$ a greater bandwidth than the actual one.

In the case of pure autoregressive models, $\mathbf{L}$ should be obtained from (11) by factoring both of the $mp \times mp$ and $m \times m$ matrices $(\mathbf{I}_p \otimes \mathbf{Q}) + \mathbf{V}_1 \boldsymbol{\Omega} \mathbf{V}_1'$ and $\mathbf{Q}$ using standard algorithms for symmetric matrices with no further special structure. Also, using (10)–(11), $|\boldsymbol{\Sigma}_\mathbf{Y}|$ is given in this case by

$$|(\mathbf{I}_p \otimes \mathbf{Q}) + \mathbf{V}_1 \boldsymbol{\Omega} \mathbf{V}_1'||\mathbf{Q}|^{n-p} = |\mathbf{L}_p|^2 |\mathbf{R}|^{2(n-p)}$$

Additionally, using certain results presented in Ansley (1979) and Mauricio (1997), it turns out that the procedure for obtaining $\mathbf{L}$ and $|\boldsymbol{\Sigma}_\mathbf{Y}| = |\mathbf{L}|^2$ described above provides a necessary (although not sufficient, expect when $q = 0$) check on the stationarity of the model. This is due to the following facts:

(i) The system of linear equations which has to be solved to obtain the autocovariance matrices appearing in (15) is singular if and only if at least one zero of $|\boldsymbol{\Phi}(B)|$ lies on the unit circle.
(ii) If $\boldsymbol{\Sigma}_\mathbf{Y}$ is not positive-definite (in which case $\mathbf{L}$ does not exist), then at least one zero of $|\boldsymbol{\Phi}(B)|$ lies outside the unit circle.

Note, however, that existence of the Cholesky decomposition does not guarantee that the model is stationary except when $q = 0$; in fact, when $q = 0$, the Cholesky factor $\mathbf{L}_p$ of $(\mathbf{I}_p \otimes \mathbf{Q}) + \mathbf{V}_1 \boldsymbol{\Omega} \mathbf{V}_1'$ exists if and only if all the zeros of $|\boldsymbol{\Phi}(B)|$ lie outside the unit circle.

Finally, to obtain the solution to $\mathbf{L}\mathbf{e} = \mathbf{y}$, with $\mathbf{y} = \mathbf{D}_{\boldsymbol{\Phi}} \tilde{\mathbf{w}}$, note that

$$\mathbf{y}_t = \tilde{\mathbf{w}}_t - \sum_{i=1}^{p} \boldsymbol{\Phi}_i \tilde{\mathbf{w}}_{t-i} \qquad \text{for } t = 1, \ldots, n$$

with $\tilde{\mathbf{w}}_j = 0$ if $j < 1$. Then, $\mathbf{e}$ can be obtained through forward substitution in $\mathbf{L}\mathbf{e} = \mathbf{y}$ without explicitly computing $\mathbf{L}^{-1}$; in fact, the recursive calculations of each row of $\mathbf{L}$ and the corresponding element of $\mathbf{e}$ can be combined in a single step as suggested by Ansley (1979). Again, noticeable computational savings can be obtained if an algorithm which takes into account the special structure of $\mathbf{L}$ is used instead of a direct implementation of the algorithm of Martin and Wilkinson (1965).

For the sake of completeness, recalling equations (13) and (14), the objective function (12) for exact maximum likelihood estimation of $\boldsymbol{\beta}$ can be written as

$$\begin{aligned} F(\boldsymbol{\beta}|\mathbf{w}) &= (\mathbf{e}'\mathbf{e})|\mathbf{L}|^{2/(mn)} \\ &= \tilde{\mathbf{e}}'\tilde{\mathbf{e}} \\ &= \sum_{t=1}^{mn} \tilde{e}_t^2, \\ \tilde{e}_t &= |\mathbf{L}|^{1/(mn)} e_t \end{aligned} \tag{17}$$

which can be minimized by using nonlinear least squares methods. Although the method for computing $\mathbf{L}$ automatically provides a necessary check for stationarity, this check is not sufficient expect when $q = 0$. Furthermore, the algorithm described in this paper can be used for either invertible or noninvertible models.

Hence, at every iteration of the algorithm used for minimizing (17), the computation of the objective function should be complemented at least with specific checks on the invertibility of the model; otherwise, the least squares algorithm might converge outside the invertibility region. Without specific checks on the stationary of mixed models, the least squares algorithm might converge (at least in theory) to a nonstationary point; although this is rare in practice, the roots of $|\mathbf{\Phi}(B)| = 0$ should be computed on convergence so as to ensure that the final estimates are admissible.

4. COMPARISON WITH EXISTING PROCEDURES

The algorithm described in Section 3 for computing $\tilde{\mathbf{w}}'\mathbf{\Sigma}_{\mathbf{W}}^{-1}\tilde{\mathbf{w}}$ and $|\mathbf{\Sigma}_{\mathbf{W}}|$ has been developed in the C programming language, along with C versions of the algorithms of Shea (1989) and Mauricio (1997); a comparison between the FORTRAN 77 versions of those two algorithms can be found in Mauricio (1997). The C code for the three algorithms has been compiled using the Borland® BCC32® compiler with no optimization options at all; the resulting programs have been executed on an IBM® compatible PC. Actual operation counts have been implemented within the C code as explained in Section 1.

To measure the relative computational efficiency of the procedure described in Section 3, the exact likelihood function has been evaluated for a variety of vector ARMA models. In Table I, the ratio between the actual numbers of floating point multiplications and divisions required by the algorithm of Shea (1989) and those required by the algorithm described in Section 3, is presented for every model considered.

It can be seen that the algorithm of Shea (1989) requires less time-consuming operations that the new algorithm only for models with q large and much greater than p. Otherwise, the new algorithm performs faster than the algorithm of Shea (1989), by a factor close to three in many cases. Also, the relative efficiency of the new algorithm increases with the dimension m of the model, whereas it does not decrease appreciably with the series length n. Thus, except for most cases with q large and much greater than p, all of the ratios are advantageous to the new algorithm, irrespective of the values of m and n.

In Table II, the ratio between the actual numbers of floating point multiplications and divisions required by the algorithm of Mauricio (1997) and those required by the algorithm described in Section 3, is presented for every model considered.

The algorithm of Mauricio (1997) requires less time-consuming operations than the new algorithm for models with q large and greater than or equal to p, the most appreciable gain occurring when q is much greater than p. Otherwise, the new algorithm performs faster than the algorithm of Mauricio (1997). The relative efficiency of the new algorithm does not decrease appreciably with the series length n, although, as opposed to the comparison with the algorithm of Shea

TABLE I

RATIOS BETWEEN THE ACTUAL NUMBERS OF TIME-CONSUMING OPERATIONS REQUIRED BY THE ALGORITHM OF SHEA (1989) AND THOSE REQUIRED BY THE NEW ALGORITHM, TO EVALUATE THE EXACT LIKELIHOOD FUNCTION FOR VARIOUS MODELS

	$m = 2$			$m = 4$		
Models	$n = 50$	$n = 100$	$n = 200$	$n = 50$	$n = 100$	$n = 200$
AR(1)	1.53	1.28	1.15	6.31	4.17	2.76
AR(2)	1.67	1.40	1.22	4.22	3.63	2.92
MA(1)	3.02	3.04	3.06	3.95	4.00	4.02
MA(2)	2.14	2.16	2.17	2.50	2.53	2.54
ARMA(1,1)	3.00	3.03	3.05	4.96	4.58	4.37
$AR(1)_4$	1.49	1.37	1.25	2.33	2.27	2.16
$MA(1)_4$	1.34	1.35	1.35	1.42	1.43	1.44
$ARMA(1,1)_4$	1.41	1.46	1.49	2.01	1.93	1.86
$AR(1)_{12}$	1.02	1.02	1.02	1.33	1.32	1.32
$MA(1)_{12}$	0.53*	0.53*	0.53*	0.51*	0.52*	0.52*
$ARMA(1,1)_{12}$	0.76*	0.74*	0.72*	1.21	1.16	1.08
$AR(1) \times MA(1)_4$	1.36	1.39	1.40	1.66	1.59	1.56
$AR(1) \times MA(1)_{12}$	0.53*	0.54*	0.54*	0.55*	0.54*	0.54*
$MA(1) \times AR(1)_4$	2.41	2.74	3.01	2.60	2.82	3.13
$MA(1) \times AR(1)_{12}$	1.14	1.30	1.56	1.35	1.37	1.42
$ARMA(1,1) \times AR(1)_4$	2.08	2.45	2.80	2.17	2.32	2.59
$ARMA(1,1) \times MA(1)_4$	1.15	1.16	1.17	1.34	1.29	1.27
$ARMA(1,1) \times AR(1)_{12}$	1.10	1.23	1.47	1.31	1.33	1.38
$ARMA(1,1) \times MA(1)_{12}$	0.49*	0.50*	0.50*	0.51*	0.50*	0.50*

Notes: An asterisk indicates that the algorithm of Shea (1989) is faster than the algorithm described in Section 3. See Hillmer and Tiao (1979) and Jenkins and Alavi (1981) for guidelines on defining and building multiplicative seasonal vector ARMA models.

(1989), here there is no clear pattern to be seen in the relative efficiency of the new algorithm with respect to the dimension m of the model. However, expect for all the cases with q large and greater than or equal to p, all of the ratios are advantageous to the new algorithm, irrespective of the values of m and n.

From the comparisons presented above, it seems clear that the new algorithm performs worse than the algorithms of Shea (1989) and Mauricio (1997) only for models with q large and, in general, greater than p. This conclusion agrees with the fact, already stated in Section 1, that the algorithm of Ansley (1979) for scalar models implies a theoretical operation count which is quadratic in q but only linear in p. Furthermore, it can be seen from both Table I and Table II that the new algorithm performs specially well for small to moderate sample sizes, where computation of the exact likelihood is better suited against any approximation.

5. CONCLUSIONS

In this paper, a method for computing the exact likelihood function of vector ARMA models has been described in detail. The algorithm has been built on

TABLE II

RATIOS BETWEEN THE ACTUAL NUMBERS OF TIME-CONSUMING OPERATIONS REQUIRED BY THE ALGORITHM OF MAURICO (1997) AND THOSE REQUIRED BY THE NEW ALGORITHM, TO EVALUATE THE EXACT LIKELIHOOD FUNCTION FOR VARIOUS MODELS

	$m = 2$			$m = 4$		
Models	$n = 50$	$n = 100$	$n = 200$	$n = 50$	$n = 100$	$n = 200$
AR(1)	1.09	1.05	1.03	1.11	1.07	1.04
AR(2)	1.14	1.08	1.05	1.07	1.06	1.04
MA(1)	1.24	1.23	1.23	1.42	1.42	1.42
MA(2)	1.16	1.14	1.13	1.23	1.21	1.20
ARMA(1,1)	1.20	1.20	1.19	1.33	1.35	1.36
$AR(1)_4$	1.19	1.15	1.10	1.05	1.05	1.05
$MA(1)_4$	0.98*	0.92*	0.89*	0.96*	0.89*	0.86*
$ARMA(1,1)_4$	0.98*	0.93*	0.90*	0.99*	0.95*	0.91*
$AR(1)_{12}$	1.23	1.22	1.20	1.04	1.04	1.04
$MA(1)_{12}$	0.89*	0.66*	0.53*	0.86*	0.62*	0.49*
$ARMA(1,1)_{12}$	0.96*	0.82*	0.68*	0.98*	0.93*	0.84*
$AR(1) \times MA(1)_4$	0.98*	0.92*	0.89*	0.96*	0.90*	0.87*
$AR(1) \times MA(1)_{12}$	0.90*	0.67*	0.54*	0.87*	0.63*	0.50*
$MA(1) \times AR(1)_4$	1.62	1.74	1.83	1.22	1.35	1.55
$MA(1) \times AR(1)_{12}$	1.34	1.43	1.59	1.06	1.08	1.12
$ARMA(1,1) \times AR(1)_4$	1.59	1.73	1.88	1.16	1.26	1.42
$ARMA(1,1) \times MA(1)_4$	0.93*	0.85*	0.80*	0.90*	0.81*	0.77*
$ARMA(1,1) \times AR(1)_{12}$	1.33	1.41	1.55	1.06	1.07	1.11
$ARMA(1,1) \times MA(1)_{12}$	0.91*	0.66*	0.53*	0.89*	0.63*	0.49*

Notes: An asterisk indicates that the algorithm of Mauricio (1997) is faster than the algorithm described in Section 3. See Hillmer and Tiao (1979) and Jenkins and Alavi (1981) for guidelines on defining and building multiplicative seasonal vector ARMA models.

the innovations method of Ansley (1979) for scalar models, an adaptation that does not seem to have been published in the past. Apart from possible minor changes through refinement in coding, the algorithm described in this article performs quite well with respect to two of the fastest methods currently available. In fact, there seems to be no computational reason for using any algorithm other than the one described in Section 3 (especially when n is not large, so that an exact computation is usually advisable), except for models with q large and greater than p; for such models, the algorithms of Shea (1989) and Mauricio (1997) give better results in terms of computational efficiency and should be used instead.

ACKNOWLEDGEMENTS

The author is grateful to Arthur B. Treadway, Rosa M. Sánchez, and an anonymous referee for some helpful suggestions. Financial support from DGICYT Project PB98-0847/98 No. 8324 (Spain) is also acknowledged.

REFERENCES

Ansley, C. F. (1979) An algorithm for the exact likelihood of a mixed autoregressive-moving average process. *Biometrika* 66, 59–65.

Box, G. E. P., Jenkins, G. M. and Reinsel, G. C. (1994) *Time Series Analysis, Forecasting and Control* (3rd Edn). Englewood Cliffs: Prentice-Hall.

Hillmer, S. and Tiao, G. (1979) Likelihood function of stationary multiple autoregressive moving average models. *Journal of the American Statistical Association* 74, 652–60.

Jenkins, G. M. and Alavi, A. S. (1981) Some aspects of modelling and forecasting multivariate time series. *Journal of Time Series Analysis* 2, 1–47.

Kohn, R. and Ansley, C. F. (1985) Computing the likelihood and its derivatives for a Gaussian ARMA model. *Journal of Statistical Computation and Simulation* 22, 229–63.

Ljung, G. M. and Box, G. E. P. (1979) The likelihood function of stationary autoregressive-moving average models. *Biometrika* 66, 265–70.

McLeod, I. (1975) Derivation of the theoretical autocovariance function of autoregressive-moving average time series. *Applied Statistics* 24, 255–6.

Martin, R. S. and Wilkinson, J. H. (1965) Symmetric decomposition of positive definite band matrices. *Numerische Mathematik* 7, 355–61.

Mauricio, J. A. (1995) Exact maximum likelihood estimation of stationary vector ARMA models. *Journal of the American Statistical Association* 90, 282–91.

——— (1997) Algorithm AS 311: The exact likelihood function of a vector autoregressive-moving average model. *Applied Statistics* 46, 157–71.

Mélard, G. (1984) Algorithm AS 197: A fast algorithm for the exact likelihood of autoregressive-moving average models. *Applied Statistics* 33, 104–14.

Pearlman, J. G. (1980) An algorithm for the exact likelihood of a high-order autoregressive-moving average process. *Biometrika* 67, 223–33.

Reinsel, G. C. (1997) *Elements of Multivariate Time Series Analysis* (2nd edn). New York: Springer-Verlag.

Shea, B. L. (1989) Algorithm AS 242: The exact likelihood of a vector autoregressive-moving average model. *Applied Statistics* 38, 161–204.

Tunnicliffe-Wilson, G. (1979) Some efficient computational procedures for high order ARMA models. *Journal of Statistical Computation and Simulation* 8, 301–9.