Statistics Canada Statistique Canada



Methodology Branch

Social Survey Methods Division

Direction de la méthodologie

Division des méthodes d'enquêtes sociales

WORKING PAPER NO. SSMD-88-022 E

METHODOLOGY BRANCH



# TIME SERIES MODELLING AND SMOOTHING METHODS FOR SAMPLE SURVEYS

SSMD-88-022E

David A. Binder

## ABSTRACT

Classical seasonal ARIMA models and their state-space representation are reviewed. The modified Kalman filter and modified fixed point smoothing algorithms using partially improper prior distributions are shown. The adaptation of these techniques to data which are subject to correlated survey error is given. We discuss likelihood maximization, smoothing methods and confidence interval estimation. Some of the algorithms needed to perform the computations are described.

**Keywords:** ARIMA models; Confidence intervals; Correlated survey errors; State-space models.

## ACKNOWLEDGEMENTS

The author is grateful to Statistics Sweden for supporting this research.

## RÉSUMÉ

Les modèles ARMMI saisonniers et leur représentation en vecteurs d'état sont révisés. Le filtre de Kalman modifié et les algorithmes modifiés de lissage pour un point fixe utilisant des distributions a priori partiellement diffuses sont indiqués. L'adaptation de ces techniques aux données qui sont exposées à l'erreur d'enquête corrélée est démontrée. Nous examinons la maximisation de la vraisemblance, des méthodes de lissage et de l'estimation par intervalle de confiance. On décrit quelques-uns des algorithmes nécessaires pour effectuer les calculs.

**Mots-clefs:** modèles ARMMI; intervalles de confiance; erreur d'enquête correlée; modèle d'état-espace

## REMERCIEMENTS

L'auteur est reconnaissant à Statistique Suède d'avoir appuyé cette recherche.

CONTENTS

## 1. Introduction

Survey organizations, both governmental and non-governmental, conduct surveys with similar data items on repeated occasions. As a result, estimates for a characteristic of interest are available over a number of time periods. This can lead to methods and analyses which are generally not available for single cross-sectional surveys.

We denote the true underlying value of a population characteristic by $\theta_t$ at time $t$. Generally, this would be a mean, proportion, total or ratio. In the case of a sample survey, the true underlying value cannot be directly observed. Instead, we have a survey estimate, $y_t$. Sometimes, we can have a vector of survey estimates, each with the same mean. For example, in the case of a rotating panel survey with no rotation group bias we have estimates $y_{1t}$, $y_{2t}$, ..., $y_{gt}$ each with mean $\theta_t$, where $g$ is the number of rotation groups. In general, we denote by $\mathbf{y}_t$ the vector of survey estimates.

Usually the survey estimates are related over time. This relationship can be separated into two main components. The component usually considered by the data producers (the survey organization) is the relationship of the sampling error, denoted by $e_t$, over time. If the $e_t$'s are correlated, then the past data can be used in the estimate for the current occasion. This can reduce the sampling error of the estimate, compared with the sampling error of the estimate which ignores the previous data.

The data users (including some users in the survey organization) are more interested, though, in the relationship of the underlying process $\{\theta_t\}$ over time. The common practice for these users is to ignore the sampling error and to fit models to the data as if these data are observed without error. In this paper we discuss a method for incorporating these survey errors into certain models. In particular, we concentrate on the case where the underlying model is a seasonal ARIMA model and the survey errors can be represented by an ARMA process up to a multiplicative factor. This is an extension of the models discussed in Binder and Dick (1988) and Binder and Hidiroglou (1988).

An additional benefit is also available to the data producers by assuming such models for the underlying process. We have pointed out that estimates can be improved by taking account of the structure of the sampling error over time. Further improvements can also be achieved by incorporating the assumptions of the underlying model for the $\theta_t$'s. We refer to this as data smoothing. However, the improvements tend to be small when the survey error is small relative to the errors of the assumed model. Therefore, such a procedure is not generally recommended, unless the survey errors are moderate, such as would be the case for small area estimation.

A general framework for this process was given by Jones (1980) as follows. Let $\theta_t = (\theta_1, \theta_2, \ldots, \theta_t)'$ be the vector of underlying population values which we want to estimate. We assume that $\theta_t$ is multivariate normal with mean $\mu_t$ and covariance $V_t$. This is the assumed model for the underlying population process. This formulation would not be appropriate in the case of non-stationary ARIMA models.

The survey observations are given by the vector $Y_t$, where

$$Y_t = X_t \theta_t + e_t \tag{1.1}$$

and $e_t$ is a multivariate normal vector of survey errors with mean zero and covariance $U_t$. The matrix, $X_t$, is usually a matrix of 0's and 1's linking the expected values of the survey estimates to the underlying population values. Here we assume that the survey samples are sufficiently large that the normal approximation to the survey sampling error can be used. The normality assumptions are not necessary though, as the resulting estimators will be minimum mean squared error if we assume the same structure for the means and covariances, without any additional distributional assumptions.

Now, using conditional arguments, the conditional expectation of $\theta_t$ given $Y_t$ is

$$E(\theta_t | Y_t) = \mu_t + (X_t' U_t^{-1} X_t + V_t^{-1})^{-1} X_t' U_t^{-1} (Y_t - X_t \mu_t), \tag{1.2}$$

with conditional variance matrix given by

$$\mathrm{Var}(\theta_t | Y_t) = (X_t' U_t^{-1} X_t + V_t^{-1})^{-1}. \tag{1.3}$$

We note that if $V_t^{-1}$ is relatively small, so that the variance of the model for $\theta_t$ is large, then (1.2) and (1.3) reduce to the minimum variance linear unbiased estimators given by Gurney and Daly (1965).

However, expressions (1.2) and (1.3) are often impractical to apply directly, since the matrices to invert have the same dimensionality as the vector $\theta_t$. Also, the matrix $V_t$ will often depend on unknown parameters which must be estimated. In this article we will assume that $\theta_t$ follows an ARIMA process with some unknown parameters. We will also assume that the survey errors can be described by an ARMA process up to a multiplicative factor. It will be assumed that the parameters of this survey error process can be estimated from the data using design-based methods. The details of this estimation will not be given here.

In Section 2 we describe how ARIMA models can be formulated using a state-space approach. This is particularly useful for formulating the likelihood function and its derivatives. In general, we use the marginal likelihood approach given by Kohn and Ansley (1986).

In Section 3 we describe our model within the state-space structure and discuss the estimation of the parameters. This is an extension of the models in Binder and Dick (1988) which consider only ARMA models. In Section 4 we detail an algorithm for performing the computations. Section 5 discusses future research.

## 2. Autoregressive Integrated Moving Average Models and its State-Space Representation

Before describing the complete model for our problem in Section 3, we review ARIMA models and a state-space representation for this model. We also review the modified Kalman filter given by Kohn and Ansley (1986) and the fixed point smoothing algorithm. In Section 3 we formulate our complete model within the state-space framework. We closely follow the formulation and the marginal likelihood approach in Ansley and Kohn (1985) and Kohn and Ansley (1986).

### 2.1 ARIMA Models

An ARMA (p,q) model for the random variables $\theta_1, \theta_2, \ldots, \theta_T$ is defined by

$$\theta_t - \alpha_1\theta_{t-1} - \cdots - \alpha_p\theta_{t-p} = \varepsilon_t - \beta_1\varepsilon_{t-1} - \cdots - \beta_q\varepsilon_{t-q}, \tag{2.1}$$

where $\{\varepsilon_t\}$ are independent $N(0,\sigma^2)$. Defining $B$ as the backshift operator, so that $B^m\theta_t = \theta_{t-m}$ and similarly $B^m\varepsilon_t = \varepsilon_{t-m}$, expression (2.1) can be written more compactly as

$$\alpha(B)\theta_t = \beta(B)\varepsilon_t, \tag{2.2}$$

where $\alpha(B) = 1 - \alpha_1 B - \ldots - \alpha_p B^p$ and $\beta(B) = 1-\beta_1 B - \ldots - \beta_q B^q$. For stationarity it is assumed that the roots of the polynomial, $\alpha(B)$, are all outside the unit circle. The ARIMA (p,d,q) model is an ARMA (p,q) model defined on $\nabla^d\theta_t$, where $\nabla = 1-B$, the differencing operator. Thus, the ARIMA (p,d,q) model is

$$\alpha(B)\nabla^d\theta_t = \beta(B)\varepsilon_t. \tag{2.3}$$

For example, for an ARIMA (1,1,1) model, expression (2.3) is $(1 - \alpha B)(1-B)\,\theta_t = (1-\beta B)\,\varepsilon_t$.

By formally multiplying out the polynomial $\alpha(B)\nabla^d$, we see that (2.3) has the same structure as (2.2) except that some roots of the resulting polynomial are on the unit circle. The seasonal ARIMA $(p,d,q)(P,D,Q)_s$ model is given by

$$\lambda(B^s)\ \alpha(B)\ \nabla_s^D\ \nabla^d\ \theta_t = \nu(B^s)\ \beta(B)\ \varepsilon_t. \tag{2.4}$$

where $\lambda(B) = 1-\lambda_1 B - \ldots - \lambda_P\ B^P$,

$\nu(B) = 1-\nu_1 B - \ldots - \nu_Q\ B^Q$ and

$\nabla_s = 1-B^s$.

The value of the seasonal factor, $s$, corresponds to the periodicity of the series; for example s=12 with monthly data, s=4 with quarterly data. For example, for an ARIMA $(1, 1, 1)(1, 1, 0)_4$ model, expression (2.4) is $(1-\lambda B^4)(1-\alpha B)(1-B^4)(1-B)\ \theta_t = (1-\beta B)\ \varepsilon_t$.

Again, we see that (2.4) has the same structural form as (2.2). The complication introduced by the non-stationarity of (2.3) or (2.4) is that we must use a modified Kalman Filter which carries a component corresponding to an improper distribution.

## 2.2 State-Space Models

We now describe a general state-space model. In Section 2.6 we show how the ARIMA model can be structured into a state-space form. In Section 3, the models we use can also be structured into the same general state-space form.

We start by defining random vectors, called state vectors, $z_0$, $z_1$, $z_2$, ..., each of dimension $r$. These state vectors are not directly observable in most cases. Instead the observations are given by

$$y_t = h_t' z_t;\ t=1, 2, \ldots, \tag{2.5}$$

where $h_t$ is a known $r$-dimensional vector. The initial conditions are that $z_0$ is multivariate normal with mean

$$\mathbf{m}(0|0) = \mathbf{m}_0(0|0), \tag{2.6}$$

and variance matrix

$$V(0|0;k) = kV_1(0|0) + V_0(0|0), \tag{2.7}$$

Without loss of generality, we will assume that $m_0(0|0) = 0$. It will be assumed that $k$ is large, so that (2.7) is the covariance matrix for a partially diffuse distribution.

The transition equation is given by

$$z_{t+1} = Fz_t + G\varepsilon_{t+1}, \tag{2.8}$$

where $F$ is an $r$ by $r$ known matrix, $G$ is an $r$ by $n$ known matrix and $\varepsilon_t$ is a multivariate normal n-dimensional vector with zero mean and diagonal covariance matrix U. Note that the models could be extended so that F and G depend on t, but we do not use this in this article.

### 2.3 Modified Kalman Filter

Because the initial conditions represent a partially diffuse distribution, the usual Kalman filter is not appropriate. See Anderson and Moore (1979) for the usual Kalman filter. We give here the modified filter as described by Ansley and Kohn (1985). We have adapted these to handle the case of a vector-values $\varepsilon_t$, rather than the one-dimensional case. Readers who wish to skip the detailed formulae may continue with Section 2.4.

We denote the conditional mean of $z_t$ given $y_1, y_2, \ldots y_\tau$ by $\mathbf{m}(t|\tau;k)$ and its conditional variance by $\mathbf{V}(t|\tau;k)$. We allow for missing $y$-values.

The recursions are given as follows. We define

$$\mathbf{m}(t+1|t;k) = \mathbf{m}_0(t+1|t) + O\ (k^{-1}), \tag{2.9}$$

where

$$\mathbf{m}_0(t+1|t) = F\mathbf{m}_0(t|t); \tag{2.10}$$

if $y_{t+1}$ is not missing, we define

$$a_{t+1} = y_{t+1} - h'_{t+1}\ \mathbf{m}_0(t+1|t) \tag{2.11}$$

and

$$\mathbf{V}(t+1|t;k) = k\mathbf{V}_1(t+1|t) + \mathbf{V}_0(t+1|t) + O\ (k^{-1}), \qquad 2.12)$$

where

$$\mathbf{V}_1(t+1|t) = F\mathbf{V}_1(t|t)\ F' \tag{2.13}$$

and

$$\mathbf{V}_0(t+1|t) = F\mathbf{V}_0(t|t)\ F' + GUG'; \tag{2.14}$$

if $y_{t+1}$ is not missing, we define

$$v_1(t+1) = h'_{t+1} V_1(t+1|t) h_{t+1} \tag{2.15}$$

and

$$v_0(t+1) = h'_{t+1} V_0(t+1|t) h_{t+1}. \tag{2.16}$$

We note that when $y_{t+1}$ is not missing, $a_{t+1}$ is a normal random variable which, conditional on $y_1, \ldots, y_t$, has mean zero and variance given by

$$kv_1(t+1) + v_0(t+1) + O(k^{-1}). \tag{2.17}$$

Finally, the updating formulas given observation $y_{t+1}$ are as follows:

$$m(t+1|t+1;k) = m_0(t+1|t+1) + O(k^{-1}) \tag{2.18}$$

and

$$V(t+1|t+1;k) = kV_1(t+1|t+1) + V_0(t+1|t+1) + O(k^{-1}), \tag{2.19}$$

where, *(i) for* $y_{t+1}$ *missing,*

$$m_0(t+1|t+1) = m_0(t+1|t), \tag{2.20}$$

$$V_1(t+1|t+1) = V_1(t+1|t), \tag{2.21}$$

$$V_0(t+1|t+1) = V_0(t+1|t); \tag{2.22}$$

*(ii) for* $y_{t+1}$ *not missing and* $v_1(t+1) = 0$,

$$m_0(t+1|t+1) = m_0(t+1|t) + V_0(t+1|t) h_{t+1} a_{t+1}/v_0(t+1), \tag{2.23}$$

$$V_1(t+1|t+1) = V_1(t+1|t),$$

$$V_0(t+1|t+1) = V_0(t+1|t) - V_0(t+1|t) h_{t+1} h'_{t+1} V_0(t+1|t)/v_0(t+1); \tag{2.24}$$

*(iii) for* $y_{t+1}$ *not missing and* $v_1(t+1) > 0$.

$$\mathbf{m}_0(t+1|t+1) = \mathbf{m}_0(t+1|t) + \mathbf{V}_1(t+1|t)\ h_{t+1}\ a_{t+1}/v_1(t+1) \tag{2.25}$$

$$\mathbf{V}_1(t+1|t+1) = \mathbf{V}_1(t+1|t) - \mathbf{V}_1(t+1|t)\ h_{t+1}\ h'_{t+1}\ \mathbf{V}_1(t+1|t)/v_1(t+1) \tag{2.26}$$

$$\begin{aligned} \mathbf{V}_0(t+1|t+1) = {} & \mathbf{V}_0(t+1|t) + \mathbf{V}_1(t+1|t)\ h_{t+1}\ h'_{t+1}\ \mathbf{V}_1(t+1|t)\ v_0(t+1)/v_1^2(t+1) \\ & - \mathbf{V}_1(t+1|t)\ h_{t+1}\ h'_{t+1}\ \mathbf{V}_0(t+1|t)/v_1(t+1) \\ & - \mathbf{V}_0(t+1|t)\ h_{t+1}\ h'_{t+1}\ \mathbf{V}_1(t+1/t)/v_1(t+1). \end{aligned} \tag{2.27}$$

For details of the proofs of these recursions, see Kohn and Ansley (1986). We note that when $y_{t+1}$ is not missing and $v_1(t+1) > 0$, the rank of $\mathbf{V}_1(t+1|t+1)$ is less than the rank of $\mathbf{V}_1(t+1|t)$, since $\mathbf{V}_1(t+1|t+1)\ h_{t+1} = 0$, but $h'_{t+1}\mathbf{V}_1(t+1|t)\ h_{t+1} \neq 0$. Therefore, if the rank of $\mathbf{V}_1(0|0)$ is R, then the rank of $\mathbf{V}_1(t|t)$ will be zero after R non-missing values. At this point, we are certain that $v_1(T) = 0$ for all $T>t$.

The recursions given by (2.9) to (2.27) yield $\mathbf{m}_0(\tau|t)$, $\mathbf{V}_1(\tau|t)$ and $\mathbf{V}_0(\tau|t)$ for $\tau=t$ or $\tau=t+1$, and $t=1, 2, \ldots, T$. These will prove useful for obtaining the marginal likelihood function in Section 2.5.

## 2.4 Modified Fixed Point Smoothing Algorithm

In Section (2.3) we obtained the conditional mean and variance of the state vector at time t given the data up to time t. For some purposes, though, we would like to have the conditional mean and variance given all the data, including observations which occur after time t. We denote this mean and variance by $\mathbf{m}(\tau|T;k)$ and $\mathbf{V}(\tau|T;k)$ for $T>\tau$. To obtain these, we apply (2.9) to (2.27) to an augmented state-space model.

In particular, we let $z_t^* = (z'_t, z'_\tau)'$,

$$F^* = \left|\begin{array}{c|c} F & 0 \\ \hline 0 & I_r \end{array}\right|, \quad G^* = \left|\begin{array}{c} G \\ \hline 0 \end{array}\right|$$

and $h_t^* = (h_t', 0')'$. Here $I_r$ is the $r$ by $r$ identity matrix. The state-space model is given by (2.5) and (2.8) where $z_t$, $F$, $G$ and $h_t$ are replaced by $z_t^*$, $F^*$, $G^*$ and $h_t^*$, respectively. We denote $\mathrm{Cov}(z_t, z_\tau | y_1, \ldots, y_s)$ by

$$C(t,\tau|s;k) = kC_1(t,\tau|s) + C_0(t,\tau|s) + O(k^{-1}). \tag{2.28}$$

The detailed recursions are given by (2.29) to (2.52). Some readers may wish to skip to Section 2.5.

$$m(\tau|\tau;k) = m_0(\tau|\tau) + O(k^{-1}) \tag{2.29}$$

$$V(\tau|\tau;k) = kV_1(\tau|\tau) + V_0(\tau|\tau) + O(k^{-1}), \tag{2.30}$$

$$C_1(\tau,\tau|\tau) = V_1(\tau|\tau) \tag{2.31}$$

and

$$C_0(\tau,\tau|\tau) = V_0(\tau|\tau), \tag{2.32}$$

where $V_1(\tau|\tau)$ and $V_0(\tau|\tau)$ are obtained from the modified Kalman filter of Section 2.3.

Now,

$$C(t+1,\tau|t;k) = kC_1(t+1,\tau|t) + C_0(t+1,\tau|t) + O(k^{-1}), \tag{2.33}$$

where

$$C_1(t+1,\tau|t) = FC_1(t,\tau|t)$$

and

$$C_0(t+1,\tau|t) = FC_0(t,\tau|t). \tag{2.34}$$

The updating equations become

$$m(\tau|t+1;k) = m_0(\tau|t+1) + O(k^{-1}) \tag{2.35}$$

$$C(t+1,\tau|t+1;k) = kC_1(t+1,\tau|t+1) + C_0(t+1,\tau|t+1) + O(k^{-1}) \tag{2.36}$$

$$V(\tau|t+1;k) = kV_1(\tau|t+1) + V_0(\tau|t+1) + O(k^{-1}) \tag{2.37}$$

where, *(i) for* $y_{t+1}$ *missing*

$$m_0(\tau|t+1) = m_0(\tau|t) \tag{2.38}$$

$$C_1(t+1,\tau|t+1) = C_1(t+1,\tau|t) \tag{2.39}$$

$$C_0(t+1,\tau|t+1) = C_0(t+1,\tau|t) \tag{2.40}$$

$$V_1(\tau|t+1) = V_1(\tau|t) \tag{2.41}$$

$$V_0(\tau|t+1) = V_0(\tau|t); \tag{2.42}$$

*(ii) for* $y_{t+1}$ *not missing and* $v_1(t+1) = 0$

$$m_0(\tau|t+1) = m_0(\tau|t) + C_0'(t+1,\tau|t)\ h_{t+1}\ a_{t+1}/v_0(t+1) \tag{2.43}$$

$$C_1(t+1,\tau|t+1) = C_1(t+1,\tau|t) \tag{2.44}$$

$$C_0(t+1,\tau|t+1) = C_0(t+1,\tau|t) - V_0(t+1|t)h_{t+1}h_{t+1}'C_0(t+1,\tau|t)/v_0(t+1) \tag{2.45}$$

$$V_1(\tau|t+1) = V_1(\tau|t) \tag{2.46}$$

$$V_0(\tau|t+1) = V_0(\tau|t) - C_0'(t+1,\tau|t)h_{t+1}h_{t+1}'C_0(t+1,\tau|t)/v_0(t+1); \tag{2.47}$$

*(iii) for* $y_{t+1}$ *not missing and* $v_1(t+1) > 0$

$$m_0(\tau|t+1) = m_0(\tau|t) + C_1'(t+1,\tau|t)\ h_{t+1}\ a_{t+1}/v_1(t+1) \tag{2.48}$$

$$C_1(t+1,\tau|t+1) = C_1(t+1,\tau|t) - V_1(t+1|t)h_{t+1}h_{t+1}'C_1(t+1,\tau|t)/v_1(t+1) \tag{2.49}$$

$$\begin{aligned} C_0(t+1,\tau|t+1) = {} & C_0(t+1,\tau|t) + V_1(t+1|t)h_{t+1}h_{t+1}'C_1(t+1,\tau|t)v_0(t+1)/v_1^2(t+1) \\ & - V_1(t+1|t)\ h_{t+1}\ h_{t+1}'\ C_0(t+1,\tau|t)/v_1(t+1) \\ & - V_0(t+1|t)\ h_{t+1}\ h_{t+1}'\ C_1(t+1,\tau|t)/v_1(t+1) \end{aligned} \tag{2.50}$$

$$V_1(\tau|t+1) = V_1(\tau|t) - C_1'(t+1,\tau|t)h_{t+1}h_{t+1}'C_1(t+1,\tau|t)/v_1(t+1) \quad (2.51)$$

$$V_0(\tau|t+1) = V_0(\tau|t) + C_1'(t+1,\tau|t)h_{t+1}h_{t+1}'C_1(t+1,\tau|t)v_0(t+1)/v_1^2(t+1)$$
$$- C_1'(t+1,\tau|t)\ h_{t+1}\ h_{t+1}'\ C_0(t+1,\tau|t)/v_1(t+1)$$
$$- C_0'(t+1,\tau|t)\ h_{t+1}\ h_{t+1}'\ C_1(t+1,\tau|t)/v_1(t+1). \quad (2.52)$$

We note that in the fixed point smoothing algorithm, if we are only interested in a linear combination of $z_\tau$, say $g'z_\tau$ for some fixed vector $g$, the computations are reduced, since we only need to carry $C_1(s,\tau|t)g$, $C_0(s,\tau|t)g$, $g'V_1(\tau|t)g$ and $g'V_0(\tau|t)g$ through the recursions, where $s=t$ or $s=t+1$. These results generalize slightly the modified fixed point smoothing algorithm in Kohn and Ansley (1986), where only $y_\tau = h_\tau' z_\tau$ for missing y-values were of interest.

## 2.5 Marginal Likelihood Function

In Section 2.2 we obtained recursions for the mean and variance of $y_t$ given the non-missing values of $y_1, \ldots, y_{t-1}$. We found that $a_t = y_t - h_t' m_0(t|t-1)$ given the non-missing values of $y_1, \ldots, y_{t-1}$ is normally distributed with mean $O(k^{-1})$ and variance $kv_1(t) + v_0(t) + O(k^{-1})$. Therefore, for any given $k$, the density function for $\{y_t\}$ is $f(y;k)$ where

$$\log f(y;k) = \sum_t [-\frac{1}{2}\log(2\pi) - \frac{1}{2}\log\{kv_1(t) + v_0(t)\}$$
$$- \frac{1}{2} a_t^2 / \{kv_1(t) + v_0(t)\}] + O(k^{-1}), \quad (2.53)$$

and the summation is taken over the non-missing y-values.

However, as $k\to\infty$ this becomes an improper density function. To remedy this, we consider a marginal density function which does not depend on $k$.

The starting conditions for the state-space model were that $z_0$ was normal with mean zero and variance $kV_1(0|0) + V_0(0|0)$, where the rank of $V_1(0|0)$ is R.

This is equivalent to assuming

$$z_0 = A\,\eta + \omega, \tag{2.54}$$

where A is an r by R fixed matrix, $\eta$ is an R-dimensional $N(0,kI_R)$ random variable, where $I_R$ is the R by R identity matrix, and $\omega$ is an r-dimensional $N(0,W)$ random variable, independent of $\eta$.

The density function for $z_0$ is

$$f(z_0;k) = (2\pi)^{-r/2}\ |kAA' + W|^{-\frac{1}{2}} \exp\{-\tfrac{1}{2}\, z_0'(kAA' + W)^{-1} z_0\}. \tag{2.55}$$

Consider now

$$\lim_{k\to\infty} k^{R/2}\ f(z_0;k). \tag{2.56}$$

We have

$$\lim_{k\to\infty} k^{R/2}\ |kAA' + W|^{-\frac{1}{2}} = |W|^{-\frac{1}{2}}\ |A'W^{-1}A|^{-\frac{1}{2}}$$

and

$$\lim_{k\to\infty} z_0'(kAA' + W)^{-1} z_0 = z_0'[W^{-1} - W^{-1}A(A'W^{-1}A)^{-1}A'W^{-1}]\ z_0. \tag{2.58}$$

Now the quadratic form given by the right hand side of (2.58) is the same as for the density of

$$[I_r - A(A'W^{-1}A)^{-1}A'W^{-1}]\ z_0, \tag{2.59}$$

which is independent of $\eta$ and independent of $A'W^{-1}z_0$. Therefore the limit of the density function in (2.56) is proportional to the singular normal density function for the random variable given by (2.59). We use this marginal density function which does not depend on the value of k. The interpretation is that our inferences are conditional on $A'W^{-1}z_0$, so that the initial condition is that $z_0$ has a singular multivariate normal distribution.

To obtain the marginal likelihood for all the data, we take

$$\lim_{k\to\infty} k^{R/2} f(y;k), \tag{2.60}$$

where $f(y;k)$ is given in (2.53), and we normalize expression (2.60) to a density function. The logarithm of the resulting density function is

$$\ell(y) = -\frac{1}{2}\sum_t \log\{2\pi v_0(t)\} - \frac{1}{2}\sum_t a_t^2/v_0(t), \tag{2.61}$$

where the summation is over the non-missing $y$-values. Suppose now that $\ell(y)$ depends on a vector of parameter $\gamma$. Taking derivatives with respect to $\gamma$, we have

$$\frac{\partial\ell}{\partial\gamma} = \sum_t [v_0(t)]^{-1} \left[\frac{1}{2}\left\{\frac{a_t^2}{v_0(t)} - 1\right\} \frac{\partial v_0(t)}{\partial\gamma} - a_t \frac{\partial a_t}{\partial\gamma}\right], \tag{2.62}$$

where, from (2.11) we have

$$\frac{\partial a_t}{\partial\gamma} = -\left|\frac{\partial m_0(t|t-1)}{\partial\gamma}\right| h_t. \tag{2.63}$$

We also have

$$-E\left|\frac{\partial^2\ell}{(\partial\gamma)(\partial\gamma)'}\right| = \frac{1}{2}\sum_t \left|v_0^2(t)\right|^{-1} \left|\left\{\frac{\partial v_0(t)}{\partial\gamma}\right\}\left\{\frac{\partial v_0(t)}{\partial\gamma}\right\}'\right|$$

$$+ \sum_t [v_0(t)]^{-1} \left\{\frac{\partial a_t}{\partial\gamma}\right\}\left\{\frac{\partial a_t}{\partial\gamma}\right\}'. \tag{2.64}$$

The maximum likelihood estimates for $\gamma$ are obtained when expression (2.62) is zero. The asymptotic variance of this estimate is given by inverting the matrix given by (2.64).

## 2.6 State-Space Representation for ARIMA Models

In Sections 2.2 to 2.5, we presented results for general state-space models. In order to implement these for our application, we show how ARIMA models presented in Section 2.1 can be represented in this form. We will also develop the initial conditions, as given by Ansley and Kohn (1985).

Consider the ARIMA $(p^*,d,q^*)(P,D,Q)_s$ given by

$$\lambda(B^s)\ \alpha^*(B)\ \nabla_s^D\ \nabla^d\ \theta_t = \nu(B^s)\ \beta^*(B)\ \varepsilon_t \tag{2.65}$$

where $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_T$ are independent $N(0,\sigma^2)$. We define $\alpha(B) = \lambda(B^s)\alpha^*(B)$ which is of degree $p = p^* + sP$.

We also define $\beta(B) = \nu(B^s)\beta^*(B)$ which is of degree $q = q^* + sQ$. We let $\Delta(B) = \nabla_s^D\nabla^d$ which is of degree $R = d + sD$. Finally we let $a^*(B) = \alpha(B)\Delta(B)$ which is of degree $S = p+R$. Therefore the model (2.65) may be written as

$$\alpha(B)\Delta(B)\theta_t = \beta(B)\varepsilon_t \tag{2.66}$$

or

$$a^*(B)\theta_t = \beta(B)\varepsilon_t \tag{2.67}$$

where

$$\alpha(B) = 1 - \alpha_1 B - \ldots - \alpha_p B^p \tag{2.68}$$

$$\Delta(B) = 1 - \Delta_1 B - \ldots - \Delta_R B^R \tag{2.69}$$

$$\beta(B) = 1 - \beta_1 B - \ldots - \beta_q B^q \tag{2.70}$$

and

$$a^*(B) = 1 - a_1^* B - \ldots - a_S^* B^S. \tag{2.71}$$

For example, for an ARIMA $(1,1,1)(1,1,0)_4$ model given by

$$(1 - \lambda B^4)(1 - \alpha^* B)(1 - B^4)(1 - B)\ \theta_t = (1 - \beta^* B)\ \varepsilon_t,$$

we have

$$\alpha(B) = 1 - \alpha^{\star}B - \lambda B^4 + \alpha^{\star}\lambda B^5.$$

$$\Delta(B) = 1 - B - B^4 + B^5$$

and $$a^{\star}(B) = 1 - (1+\alpha^{\star})B + \alpha^{\star}B^2 - (1+\lambda)B^4 + (1+\alpha^{\star})(1+\lambda)B^5 - \alpha^{\star}(1+\lambda)B^6 + \lambda B^8 - \lambda(1+\alpha^{\star})B^9 + \alpha^{\star}\lambda B^{10}.$$

Note that $p=5$, $q=1$, $R=5$ and $S=10$.

We now define $z_t = (z_{1t}, \ldots, z_{rt})'$, the state vector. Let $r = \max(S,q+1)$. If $S<q+1$, we define $a^{\star}_{S+1} = \ldots = a^{\star}_r = 0$. If $S>q+1$, we define $\beta_{q+1} = \ldots = \beta_{r-1} = 0$. We let

$$z_t = A_1 \begin{vmatrix} \theta_t \\ \theta_{t-1} \\ \vdots \\ \theta_{t-r+1} \end{vmatrix} + A_2 \begin{vmatrix} \varepsilon_t \\ \varepsilon_{t-1} \\ \vdots \\ \varepsilon_{t-r+2} \end{vmatrix}, \tag{2.72}$$

where

$$A_1 = \begin{vmatrix} 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & a^{\star}_2 & a^{\star}_3 & \ldots & a^{\star}_{r-1} & a^{\star}_r \\ 0 & a^{\star}_3 & a^{\star}_4 & \ldots & a^{\star}_r & 0 \\ \vdots & & & & & \\ 0 & a^{\star}_r & 0 & \ldots & 0 & 0 \end{vmatrix}, \tag{2.73}$$

an $r$ by $[\max(p,1)+R]$ matrix, and

$$A_2 = \begin{vmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ -\beta_1 & -\beta_2 & -\beta_3 & \dots & -\beta_{r-2} & -\beta_{r-1} \\ -\beta_2 & -\beta_3 & -\beta_4 & \dots & -\beta_{r-1} & 0 \\ \vdots & & & & & \\ -\beta_{r-1} & 0 & 0 & \dots & 0 & 0 \end{vmatrix}, \tag{2.74}$$

an $r$ by $(r-1)$ matrix.

Now for $h_t = (1,0, \dots 0)'$,

$$F = \begin{vmatrix} a_1^* & 1 & 0 & \dots & 0 \\ a_2^* & 0 & 1 & \dots & 0 \\ \vdots & & & & \\ a_{r-1}^* & 0 & 0 & \dots & 1 \\ a_r^* & 0 & 0 & \dots & 0 \end{vmatrix} \tag{2.75}$$

and $G = (1, -\beta_1, -\beta_2, \dots, -\beta_{r-1})'$ we have $\theta_t = h_t' z_t$ satisfies model (2.67) when the state vectors given by (2.72) satisfy the transition equation (2.8). This representation for model (2.67) was given by Harvey and Phillips (1979).

To complete the specification of the state-space formulation, we need initial conditions. Taking model (2.66), we let $\Delta(B)\theta_t = u_t$, so that

$$\alpha(B)u_t = \beta(B)\varepsilon_t. \tag{2.76}$$

We assume that this ARMA $(p,q)$ model for $\{u_t\}$ is stationary. The following, given by Ansley and Kohn (1985), specifies the initial conditions for $z_0$. Note that $m_0(0|0) = 0$, so we need to specify $V_1(0|0)$ and $V_0(0|0)$ of (2.7).

Consider the vector $\theta_- = (\theta_0, \theta_{-1}, \dots, \theta_{-S+1})'$. Let $\eta = (\theta_{-p}, \theta_{-p-1}, \dots, \theta_{-S+1})'$. We denote $u_- = (u_0, u_{-1}, \dots, u_{-p+1})'$. We assume $\eta$ is $N(0, kI_R)$ and $u_-$ is $N(0, \sigma^2 V_u)$, independent of $\eta$.

Expressing $\eta$ and $u_-$ as a function of $\theta_-$, we obtain

$$\begin{vmatrix} u_- \\ \eta \end{vmatrix} = M\,\theta_- \tag{2.77}$$

where $M$ is a $[\max(p,1) + R]$ - square matrix. This matrix is the identity matrix if $R=0$; otherwise it is

$$M = \left|\begin{array}{ccccccccc} 1 & -\Delta_1 & -\Delta_2 & \cdots & -\Delta_R & 0 & & \cdots & 0 \\ 0 & 1 & -\Delta_1 & \cdots & -\Delta_{R-1} & -\Delta_R & & \cdots & 0 \\ \vdots & & & & & & & & \\ 0 & 0 & 0 & \cdots & -\Delta_{R-p+1} & -\Delta_{R-p+2} & & \cdots & -\Delta_R \\ \hline & & \mathbf{0} & & & \multicolumn{4}{|c}{I_R} \end{array}\right|. \tag{2.78}$$

For example, when $\Delta(B) = (1-B^4)(1-B)$, and $p=5$ as in the ARIMA $(1,1,1)(1,1,0)_4$ example, we have

$$M = \left|\begin{array}{ccccc|ccccc} 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & -1 & 1 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right|.$$

Therefore,

$$\theta_- = M^{-1} \begin{vmatrix} u_- \\ \eta \end{vmatrix}. \tag{2.79}$$

Since
$$z_0 = A_1\,\theta_- + A_2\,\varepsilon_-, \tag{2.80}$$

where $\varepsilon_- = (\varepsilon_0, \varepsilon_{-1}, \ldots, \varepsilon_{-r+2})'$ if $r \geq 2$ and $\varepsilon_-$ is the null vector if $r=1$, we have

$$z_0 = A_1 M^{-1} \begin{vmatrix} u_- \\ \eta \end{vmatrix} + A_2 \varepsilon_- . \qquad (2.81)$$

This may be written as

$$z_0 = K_1 u_- + K_2 \eta + A_2 \varepsilon_- , \qquad (2.82)$$

where $K_1$ is the first $\max(p,1)$ columns of $A_1 M^{-1}$ and $K_2$ is the last $R$ columns of $A_1 M^{-1}$. Therefore, the variance of $z_0$ is $kV_1(0|0) + V_0(0|0)$, where

$$V_1(0|0) = K_2 K_2' \qquad (2.83)$$

and

$$V_0(0|0) = \sigma^2 (K_1 V_u K_1' + K_1 C_{UE} A_2' + A_2 C_{UE}' K_1' + A_2 A_2'), \qquad (2.84)$$

where the matrix $C_{UE}$ is the covariance between $u_-$ and $\varepsilon_-$ and $\sigma^2 V_u$ is the covariance matrix for $u_-$.

In Section 4 we describe a method for obtaining $V_u$ and $C_{UE}$.

A simple example is given now to show how these computations are carried out. Consider the ARIMA (1,1,1) model given by

$$(1 - \alpha^* B)(1-B)\, \theta_t = (1-\beta^* B)\varepsilon_t$$

Therefore,
$$\alpha(B) = 1 - \alpha^* B$$
$$\Delta(B) = 1-B$$
$$\beta(B) = 1-\beta^* B$$
$$a^*(B) = 1 - (1 + \alpha^*)B + \alpha^* B^2,$$

so that $p=1$, $q=1$, $R=1$, $S=2$ and $r=2$. We have

$$A_1 = \begin{vmatrix} 1 & 0 \\ 0 & -\alpha^* \end{vmatrix}$$

$$A_2 = \begin{vmatrix} 0 \\ -\beta \end{vmatrix}$$

$$F = \begin{vmatrix} 1 + \alpha^* & 1 \\ -\alpha^* & 0 \end{vmatrix}$$

$$G = \begin{vmatrix} 1 \\ -\beta^* \end{vmatrix}$$

and

$$M = \begin{vmatrix} 1 & -1 \\ 0 & 1 \end{vmatrix}.$$

Therefore,

$$A_1 M^{-1} = \begin{vmatrix} 1 & 1 \\ 0 & -\alpha^* \end{vmatrix},$$

so that

$$K_1 = \begin{vmatrix} 1 \\ 0 \end{vmatrix} \quad \text{and} \quad K_2 = \begin{vmatrix} 1 \\ -\alpha^* \end{vmatrix}$$

Using the methods described in Section 4, we find

$$C_{UE} = [\ 1\ ]$$

and

$$V_u = [\ v\ ],$$

where $v = (1 - 2\alpha^{*}\beta^{*})/\{1-(\alpha^{*})^{2}\}$.

Therefore,

$$V_1(0|0) = \begin{vmatrix} 1 & -\alpha^{*} \\ -\alpha^{*} & (\alpha^{*})^{2} \end{vmatrix}$$

$$V_0(0|0) = \sigma^{2} \begin{vmatrix} v & -\beta \\ -\beta & \beta^{2} \end{vmatrix}.$$

This completes the specification of the ARIMA models in state-space form. In the next section we show how this can be extended to the case where the observations are subject to survey error.

## 3. ARIMA Models with Observations Subject to Survey Error

### 3.1 ARMA Models for Survey Error

When a time series $\{\theta_t\}$ which follows an ARIMA process is observed exactly, the likelihood function for the unknown parameters can be derived using the state-space formulation given in Section 2. Recursive relations for the derivatives of the likelihood function can also be obtained using methods given in Section 4.

However, when the observed time series is the result of a series of sample surveys, the survey sampling error should be taken into account when deriving the likelihood function. The actual structure of the survey error will depend on the sample design and the population characteristics. We let $y_t = \theta_t + e_t$ for $t=1, \ldots, T$ be the observed series where $e_t$ is the survey sampling error. The simplest case is where the surveys are non-overlapping with small sampling fractions so that the $e_t$'s are approximately independent.

In a rotating panel survey, the $e_t$'s will be correlated. Suppose there are q panels and one is dropped and replaced by a new independent panel on each occasion. The panels rotate so that an entering panel leaves the survey after q time periods. Assuming small sampling fractions, this implies that the correlation between $e_t$ and $e_s$ is zero for $s > t + q$. If the correlations are constant, this implies that $k_t e_t$ is a pure moving average process, ARMA (0,q). Here, $k_t$ can vary to reflect different variances for each point in time, although the autocorrelations are assumed constant.

If on each occasion a random set of units is dropped, it may be reasonable to assume that $e_t$, or at least a multiple of $e_t$, given by $k_t e_t$, is first order autogressive, ARMA (1,0). This implies that the correlation between $k_t e_t$ and $k_{t+s} e_{t+s}$ is $\alpha^s$ for some $\alpha$.

We see, therefore, that it can often be assumed that $k_t e_t$ is an ARMA process. It may be possible to assume other structures which admit a state-space form and what follows could be modified to satisfy that structure. We also assume that the parameters of the state-space model can be estimated using design-based methods. This is not necessarily straight-forward in general, and more research into estimating these parameters is needed. However, here we assume that these parameters are known.

### 3.2 The Data Model

The complete model we wish to consider, therefore, is the case where $\{\theta_t\}$ is an ARIMA process and the survey errors, $\{e_t\}$, follow an ARMA process. Using the modified Kalman filters, we will develop the marginal likelihood function. Maximizing this function with respect to the unknown parameters yields parameter estimates. In this way, we can estimate the parameters of an ARIMA model in the presence of survey errors.

In traditional ARIMA modelling with no survey error, the series is differenced using $\Delta(B)$ so that the derived series is a stationary ARMA process. However, in our application, differencing the survey estimates would complicate the covariance structure of the survey errors. The approach given here is easier to implement and missing $y$-values can be handled within the same framework. In Section 4 we introduce regression terms into the model as well.

As in Section 2, we let $\theta_t$ can be described by the ARIMA model:

$$\alpha(B)\Delta(B)\theta_t = \beta(B)\varepsilon_t, \tag{3.1}$$

where

$$\alpha(B) = 1 - \alpha_1 B - \dots - \alpha_p B^p, \tag{3.2}$$

$$\beta(B) = 1 - \beta_1 B - \dots - \beta_1 B^q \tag{3.3}$$

$$\Delta(B) = 1 - \Delta_1 B - \dots - \Delta_R B^R \tag{3.4}$$

and the $\varepsilon_t$'s are independent $N(0,\sigma^2)$.

We now assume $k_t\ e_t$ follows an ARMA $(m,n)$ model

$$\phi(B)(k_t e_t) = \Psi(B)\eta_t$$

where

$$\phi(B) = 1 - \phi_1 B - \dots - \phi_m B^m \tag{3.5}$$

$$\Psi(B) = 1 - \Psi_1 B - \dots - \Psi_n B^n \tag{3.6}$$

and the $\eta_t$'s are $N(0,\tau^2)$. The observations are given by

$$y_t = \theta_t + e_t, \quad \text{for} \quad t=1, \dots, T. \tag{3.7}$$

This model can now be put into state-space form.
We let

$$a^*(B) = \alpha(B)\Delta(B)$$
$$= 1 - a_1^* B - \dots - a_S^* B^S, \tag{3.8}$$

where $S = p+R$. We let $r_1 = \max(S, q+1)$, $r_2 = \max(m, n+1)$ and $r = r_1+r_2$. We let $h_1 = (1, 0, \dots, 0)'$ be an $r_1$-dimensional vector, $h_{2t} = (k_t^{-1}, 0, \dots, 0)'$ be an $r_2$-dimensional vector and $h_t = (h_1', h_{2t}')'$. We let $F_1$ be the $r_1$ by $r_1$ matrix given by

$$F_1 = \begin{vmatrix} a_1^* & 1 & 0 & \dots & 0 \\ a_2^* & 0 & 1 & \dots & 0 \\ \vdots & & & & \\ a_{r_1-1}^* & 0 & 0 & \dots & 1 \\ a_{r_1}^* & 0 & 0 & \dots & 0 \end{vmatrix}, \tag{3.9}$$

where, if $S<r_1$, then $a^*_{S+1} = \dots = a^*_{r_1} = 0$. We let $F_2$ be the $r_2$ by $r_2$ matrix given by

$$F_2 = \begin{vmatrix} \phi_1 & 1 & 0 & \dots & 0 \\ \phi_2 & 0 & 1 & \dots & 0 \\ \vdots & & & & \\ \phi_{r_2-1} & 0 & 0 & \dots & 0 \\ \phi_{r_2} & 0 & 0 & \dots & 0 \end{vmatrix}, \tag{3.10}$$

where, if $m<r_2$, then $\phi_{m+1} = \dots = \phi_{r_2} = 0$. The $r \times r$ matrix $F$ is given by

$$F = \begin{vmatrix} F_1 & 0 \\ 0 & F_2 \end{vmatrix}. \tag{3.11}$$

We let $G_1$ by the $r_1$-dimensional vector given by
$G_1 = (1, -\beta_1, \dots, -\beta_{r_1-1})'$, where, if $q < r_1-1$, then
$\beta_{q+1} = \dots = \beta_{r_1-1} = 0$. We let $G_2$ be the $r_2$-dimensional vector given by
$G_2 = (1, -\psi_1, \dots, -\psi_{r_2-1})'$, where, if $n < r_2-1$, then
$\psi_{n+1} = \dots = \psi_{r_2-1} = 0$. The matrix $G$ is an $r$ by 2 matrix given by

$$G = \begin{vmatrix} G_1 & 0 \\ 0 & G_2 \end{vmatrix}. \tag{3.12}$$

Finally, letting

$$U = \begin{vmatrix} \sigma^2 & 0 \\ 0 & \tau^2 \end{vmatrix}. \tag{3.13}$$

completes the specification of the transition equation (2.8).

For the initial state vector, we let $\mathbf{m}_0(0|0) = 0$. We let

$$V_1(0|0) = \begin{vmatrix} V_{1,1}(0|0) & 0 \\ 0 & 0 \end{vmatrix} \qquad 3.14)$$

where $V_{1,1}(0|0)$ is an $r_1$ by $r_1$ matrix derived analogously to (2.83). We let

$$V_0(0|0) = \begin{vmatrix} V_{1,0}(0|0) & 0 \\ 0 & V_{2,0}(0|0) \end{vmatrix}, \qquad 3.15)$$

where $V_{1,0}(0|0)$ is an $r_1$ by $r_1$ matrix derived analogously to (2.84), and $V_{2,0}(0|0)$ is also derived analogously to (2.84), using the parameters $\phi_1, \ldots, \phi_m$, $\psi_1, \ldots, \psi_n$ and $\tau^2$.

This completes the specification of the data model in state-space form. From this, using the modified Kalman Filter, the marginal likelihood function given by (2.61) can be derived.

### 3.3 Data Smoothing

Our observations consist of $y_t = \theta_t + e_t$, where $e_t$ is the survey sampling error, for $t=1, \ldots, T$. The population characteristics of interest are $\theta_1, \ldots, \theta_T$. Once all the parameters of the state-space model have been estimated, we can use the modified fixed point smoothing algorithm to obtain $E(\theta_\tau | y_1, \ldots, y_T)$, for $\tau<T$.

In particular, for the state space model of Section 3.2, with state vectors $z_1, \ldots, z_T$, we have $\theta_\tau = g' z_\tau$, where $g = (h_1', 0, \ldots, 0)'$. Using the modified fixed point smoothing algorithm of Section 2.4, we can obtain

$$E(\theta_\tau | y_1, \ldots, y_T; k) = g' m_0(\tau|T) + O(k^{-1}) \tag{3.16}$$

and

$$V(\theta_\tau | y_1, \ldots, y_T; k) = kg' V_1(\tau|T)g + g' V_0(\tau|T)g + O(k^{-1}). \tag{3.17}$$

We now extend this to the estimation of change, $\theta_\tau - \theta_\omega$ for $\tau > \omega$. From Section 2.4, starting at $t=\omega$ and continuing to $t=\tau$, we obtain

$$E(\theta_\tau - \theta_\omega | y_1, \ldots, y_\tau; k) = g' m_0(\tau|\tau) - g' m_0(\omega|\tau) + O(k^{-1}), \tag{3.18}$$

$$\begin{aligned} Var(\theta_\tau - \theta_\omega | y_1, \ldots, y_\tau; k) &= k[g' V_1(\tau|\tau)g + g' V_1(\omega|\tau)g - 2g' C_1(\tau,\omega|\tau)g] \\ &\quad + g' V_0(\tau|\tau)g + g' V_0(\omega|\tau)g - 2g' C_0(\tau,\omega|\tau)g + O(k^{-1}) \end{aligned} \tag{3.19}$$

and $Cov(z_\tau, \theta_\tau - \theta_\omega | y_1, \ldots, y_\tau; k)$

$$\begin{aligned} &= k[V_1(\tau|\tau)g - C_1(\tau,\omega|\tau)g] \\ &\quad + V_0(\tau|\tau)g - C_0(\tau,\omega|\rho)g + O(k^{-1}). \end{aligned} \tag{3.20}$$

The quantities given by expressions (3.18) to (3.20) can then be used in the modified fixed point smoothing algorithm, using $\theta_\tau - \theta_\omega$ as the fixed point, so that the state-space model is

$$\begin{vmatrix} z_{t+1} \\ \theta_\tau - \theta_\omega \end{vmatrix} = \begin{vmatrix} F & 0 \\ 0 & 1 \end{vmatrix} \begin{vmatrix} z_t \\ \theta_\tau - \theta_\omega \end{vmatrix} + \begin{vmatrix} G \\ 0 \end{vmatrix} \begin{vmatrix} \varepsilon_{t+1} \\ \eta_{t+1} \end{vmatrix}, \tag{3.21}$$

with observations

$$y_{t+1} = (h_{t+1}', 0) \begin{vmatrix} z_{t+1} \\ \theta_\tau - \theta_\omega \end{vmatrix}, \tag{3.22}$$

for $t = \tau, \tau+1, \ldots, T-1$.

This procedure could be generalized to obtain the conditional mean and variance of any linear combination $\ell_1\theta_1 + \ell_2\ \theta_2 + ... + \ell_T\theta_T$ for fixed values of $\ell_1, \ldots, \ell_T$. In this case the state-vector used in place of (3.21) is $(z'_{t+1}, \ell_1\ \theta_1 + \ldots + \ell_t\theta_t)'$ and the last row and column of the conditional variance matrix must be updated analogously to expressions (3.19) and (3.20).

### 3.4 Confidence Intervals

In the model of Section 3.2, the unknown parameters are $\alpha_1, \ldots, \alpha_p, \beta_1, \ldots, \beta_q$ and $\sigma^2$. In fact, for the more general seasonal model given by

$$\lambda(B^S)\ \alpha(B)\ \Delta(B)\ \theta_t = \nu(B^S)\ \beta(B)\ \varepsilon_t, \tag{3.23}$$

where
$$\lambda(B) = 1 - \lambda_1 B - \ldots - \lambda_P B^P \tag{3.24}$$

$$\alpha(B) = 1 - \alpha_1 B - \ldots - \alpha_p B^p \tag{3.25}$$

$$\nu(B) = 1 - \nu_1 B - \ldots - \nu_Q B^Q \tag{3.26}$$

$$\beta(B) = 1 - \beta_1 B - \ldots - \beta_q B^q \tag{3.27}$$

and
$$\Delta(B) = \nabla_S^D\ \nabla^d, \tag{3.28}$$

the unknown parameters are $\gamma' = (\lambda_1, \ldots, \lambda_P, \alpha_1, \ldots, \alpha_p, \nu_1, \ldots, \nu_Q, \beta_1, \ldots, \beta_q, \sigma^2)$.

To obtain the maximum likelihood estimates, $\hat{\gamma}$, for the parameters, $\gamma$, it is necessary to solve the likelihood equations given by (2.62). Asymptotically, $\hat{\gamma} - \gamma$ will be approximately multivariate normal with covariance matrix, $V_r$, given by inverting the matrix given by (2.64). By substituting parameters estimates into (2.64), we can obtain confidence intervals for components of $\gamma$. Hypothesis testing can also be performed.

In order to obtain the derivative of the likelihood function and the Fisher information matrix, given by expressions (2.62) and (2.64), it is necessary to compute $\partial a_t/\partial\gamma$ and $\partial v_0(t)/\partial\gamma$. Since $a_t$ and $v_0(t)$ are obtained recursively from the modified Kalman filter, these same recursions can be used to obtain the required derivatives. For example, expression (2.14) is

$$V_0(t+1|t) = FV_0(t|t)F' + GUG'.$$

Differentiating with respect to $\alpha_i$ gives

$$\frac{\partial V_0(t+1|t)}{\partial \alpha_i} = \left| \frac{\partial F}{\partial \alpha_i} \right| V_0(t|t)F' + F \left| \frac{\partial V_0(t|t)}{\partial \alpha_i} \right| F'$$

$$+ FV_0(t|t) \left| \frac{\partial F}{\partial \alpha_i} \right|', \tag{3.29}$$

since $\partial G/\partial \alpha_i = 0$ and $\partial U/\partial \alpha_i = 0$.

In addition to the confidence intervals for the unknown parameters, we also would like to have a confidence interval for our estimate of $\theta_\tau$ given $y_1, \ldots, y_T$. If all of the parameters are known, we have the variance given by $g'V_0(\tau|T)g$, where $\theta_\tau = g'z_\tau$. This assumes that $v_1(T) = 0$. However, this does not include the sampling variance due to estimating the parameters, $\gamma$.

Denoting by $\hat{m}_0(\tau|T)$ the estimate of $m_0(\tau|T)$ at $\gamma = \hat{\gamma}$, we take a Taylor series expansion of $\hat{m}_0(\tau|T)$ to obtain

$$\hat{m}_0(\tau|T) \doteq m_0(t|T) + \left| \frac{\partial m_0(\tau|T)}{\partial \gamma} \right| (\hat{\gamma}-\gamma) + o \parallel \hat{\gamma}-\gamma \parallel \tag{3.30}$$

Since $\hat{\gamma}$ is a consistent estimator for $\gamma$, we have

$$\theta_\tau - g'm_0(\tau|T) \doteq$$

$$g'[z_\tau - m_0(\tau|T)] - g' \left| \frac{\partial m_0(\tau|T)}{\partial \gamma} \right| (\hat{\gamma}-\gamma) + o \parallel \hat{\gamma}-\gamma \parallel . \tag{3.31}$$

Therefore

$$E[\{\theta_\tau - g'\hat{m}_0(\tau|T)\}^2]$$

$$\doteq g'V_0(\tau|T)g + g'\left|\frac{\partial m_0(\tau|T)}{\partial\gamma}\right| V_\Gamma \left|\frac{\partial m_0(\tau|T)}{\partial\gamma}\right|' g. \quad (3.32)$$

To estimate $g'[\partial m_0(\tau|T/\partial\gamma]$, we use the recursions given in the modified fixed point smoothing algorithm to obtain the required derivatives.

### 3.5 Likelihood Maximization

In Section 4 we provide some details for the computation of the marginal likelihood function and its derivatives with respect to the unknown parameters, $\gamma$. From this we can compute $\ell(y;\gamma)$, the logarithm of the marginal likelihood function, given by (2.61), as well as $\partial\ell(y;\gamma)/\partial\gamma$ and

$$J = -E\left|\frac{\partial^2\ell}{(\partial\gamma)(\partial\gamma')}\right|. \quad (3.33)$$

A number of routines for maximizing a function are possible. We suggest the Davidon-Fletcher-Power method, described in, for example, Dennis and Schnabel (1983). Assume that $\gamma$ is a $c$-dimensional vector. For example, for model (3.23), $c = P+Q+p+q+1$. The algorithm is now described.

STEP 1: Start with an initial value, $\gamma^{(0)}$. See Note 1 below.

Let $g_0 = \partial\ell(y;\ \gamma^{(0)})/\partial\gamma$.

STEP 2: Let $H^{(0)} = -J$, where $J$ as given by (3.34) is computed at $\gamma^{(0)}$.

STEP 3: Perform steps 4 to 6 for $i = 1, 2, \ldots, c+1$.

STEP 4: Compute $\delta_i = -H^{(i-1)}\, g^{(i-1)}$.

STEP 5: Compute $m_i$ to maximize $\ell(y;\gamma^{(i-1)} + m_i\ \delta_i)$, where $m_i$ is a scalar. See Note 2 below. Set

$$\gamma^{(i)} = \gamma^{(i-1)} + m_i \delta_i$$

and

$$g_i = \frac{\partial \ell(y;\gamma^{(i)})}{\partial \gamma}.$$

STEP 6: Test for convergence. See Note 3 below. If convergence has been achieved, end the algorithm.

STEP 7: Perform Steps 8 and 9 for $i=1, \ldots, c$. If $i = c+1$, let $\gamma^{(0)} = \gamma^{(c+1)}$, $g_0 = g_{c+1}$ and go to Step 2.

STEP 8: Let $\xi_i = g_i - g_{i-1}$.

STEP 9: Compute

$$H^{(i)} = H^{(i-1)} + m_i \frac{\delta_i\ \delta_i'}{g_{i-1}'\ H^{(i-1)}\ g_{i-1}} - \frac{H^{(i-1)}\ \xi_i\ \xi_i'\ H^{(i-1)}}{\xi_i'\ H^{(i-1)}\ \xi_i}.$$

Go to Step 3.

**Note 1**

Starting points can often be difficult. For a problem where $c$ is large, it may help to reduce the dimensionality of the problem by setting some of the higher-order autogressive or moving average parameters to zero. This is known as masking. Then after convergence (with a weaker convergence criterion), restart with more dimensions, where the starting values are zero for the previously masked parameters, and using the converged values from the previous iterations.

**Note 2**

Step 5 is a one dimensional maximization problem. Ideally, if $H^{(i-1)}$ is the true Hessian matrix and the function is quadratic, then the function is maximized at $m_i$=1. We suggest the following procedure. Compute the function at $m_i^{(0)} = 1$, $m_i^{(1)} = km_i^{(0)}$ and at $m_i^{(2)} = m_i^{(2)}/k$ where k is 2/3, say. Let the function values be $f_0$, $f_1$ and $f_2$ respectively. Also let $f^*$ be the function value at $m_i = 0$.

Case (i):

When $f_0$ is the largest of $f_0$, $f_1$, $f_2$ and $f^*$, fit a quadratic equation through $m_i^{(0)}$, $m_i^{(1)}$ and $m_i^{(2)}$; maximize that quadratic at $m_i^{(3)}$ and compute the function value, $f_3$ at $m_i^{(3)}$. Use $m_i$ to be that value for which the maximum of $f_0$ and $f_3$ is achieved.

Case (ii):

When $f_1$ is the largest of $f_0$, $f_1$, $f_2$ and $f^*$, perform the following in order:

1. Set $m_i^{(2)}$ to $m_i^{(0)}$ and $f_2$ to $f_0$.
2. Set $m_i^{(0)}$ to $m_i^{(1)}$ and $f_0$ to $f_1$.
3. Compute $m_i^{(1)} = km_i^{(0)}$, compute its function value which is defined as the new $f_1$ and check which case now occurs.

Case (iii):

When $f_2$ is the largest of $f_0$, $f_1$, $f_2$ and $f^*$, perform the following in order:

1. Set $m_i^{(1)}$ to $m_i^{(0)}$ and $f_1$ to $f_0$.
2. Set $m_i^{(0)}$ to $m_i^{(2)}$ and $f_0$ to $f_2$.
3. Compute $m_i^{(2)} = m_i^{(0)}/k$, compute its function value which is defined as the new $f_2$ and check which case now occurs.

**Case (iv):**

When $f^*$ is the largest of $f_0$, $f_1$, $f_2$ and $f^*$, perform the following in order:

1. Set $m_i^{(2)}$ to $m_i^{(1)}$ and $f_2$ to $f_1$.

2. Compute $m_i^{(0)} = km_i^{(2)}$ and set $f_0$ to its function value.

3. Compute $m_i^{(1)} = km_i^{(0)}$, compute its function value which is defined as the new $f_1$ and check which case now occurs.

When setting the $m_i$-values, checks should be made to ensure that the $m_i$-values are not too large so as to overstep the parameter space. Useful checks are $|\lambda_i|<1$, $|\alpha_i|<1$, $|\nu_i|<1$, $|\beta_i|<1$ and $\sigma^2 > 0$.

Repeat the procedure until Case (i) occurs.

The maximum of the quadratic function through $m_i^{(0)}$, $m_i^{(1)}$ and $m_i^{(2)}$ is given by

$$m_i^{(3)} = m_i^{(0)} + \frac{(f_0-f_1)(m_i^{(2)}-m_i^{(0)})^2 - (f_0-f_2)(m_i^{(0)}-m_i^{(1)})^2}{2[(f_0-f_1)(m_i^{(2)}-m_i^{(0)}) + (f_0-f_2)(m_i^{(0)}-m_i^{(1)})]}. \quad (3.34)$$

**Note 3**

A number of tests for convergence are available. We suggest the following. We denote $g_i = (g_{i1}, \ldots, g_{ic})'$ and $\delta_i = (\delta_{i1}, \ldots, \delta_{ic})'$.

The procedure is deemed to have converged if one of the following occurs:

$$\max_j\left\{\frac{|g_{ij}\,\gamma_j^{(i)}|}{\ell(y;\gamma^{(i)})}\right\} \text{ is small, say } 10^{-7}$$

$$\text{or } \max_j\left\{\frac{m_i|\delta_{ij}|}{|\gamma_j^{(i)}|}\right\} \text{ is small, say } 10^{-7}.$$

The program should also abort if there have been a large number of iterations, say more than 100c.

## 4. Computations

### 4.1 The Model

In this section we give the detailed algorithms to compute the marginal likelihood function and its derivatives with respect to the parameters. The assumed model for the observations, $y_1, \ldots, y_T$, is as follows:

$$\lambda(B^S)\ \alpha^*(B)\ \nabla_S^D\ \nabla^d\ \theta_t = \nu(B^S)\quad \beta^*(B)_t\varepsilon \tag{4.1}$$

$$\phi(B)(k_t e_t) = \psi(B)\ \eta_t \tag{4.2}$$

$$y_t = \theta_t + e_t, \tag{4.3}$$

where

$$\lambda(B) = 1 - \lambda_1 B - \ldots - \lambda_P B^P \tag{4.4}$$

$$\alpha^*(B) = 1 - \alpha_1^* B - \ldots - \alpha_p^* B^p \tag{4.5}$$

$$\nu(B) = 1 - \nu_1 B - \ldots - \nu_Q B^Q \tag{4.6}$$

$$\beta^*(B) = 1 - \beta_1^* B - \ldots - \beta_q^* B^q \tag{4.7}$$

$$\phi(B) = 1 - \phi_1 B - \ldots - \phi_m B^m \tag{4.8}$$

$$\psi(B) = 1 - \psi_1 B - \ldots - \psi_n B^n, \tag{4.9}$$

$\{\varepsilon_t\}$ are independent $N(0,\sigma^2)$, $\{\eta_t\}$ are independent $N(0,\tau^2)$ and $\{\varepsilon_t\}$ and $\{\eta_t\}$ are independent of each other. For further generality, we will also add a regression component to the observations, so that

$$y_t = x_t' b + \theta_t + e_t, \tag{4.10}$$

where $b$ is an $L$ by 1 vector.

The following are assumed known: $\{\phi_i\}$, $\{\psi_i\}$, $\{k_t\}$, $\{x_t\}$, and $\tau^2$. Missing y-values are permitted.

The regression component in (4.10) can be handled two ways. One way would be to let $\{y_t - x_t' b\}$ be our observations in the likelihood function. When maximizing the likelihood, we would need to add a term for $\partial a_t/\partial b$ in (2.62). An alternative would be to add $b$ to the state vector. We would have

$$b_{t+1} = b_t. \tag{4.11}$$

The advantage of this approach is that the state-space formulation can be modified to also include stochastic regression coefficients, so that the transition equation becomes

$$b_{t+1} = b_t + H\,\xi_{t+1}, \tag{4.12}$$

where $\xi_{t+1}$ is multivariate normal with mean zero and a diagonal covariance matrix. We do not pursue this here.

For model (4.11), where the first $L$ components of the state vector correspond to $b$, we have $h_t$ in Section 3.2 replaced by

$$(x_t', h_1', h_{2t}')'. \tag{4.13}$$

The initial conditions are the $\mathrm{Var}(b) = kI$, so that the modified Kalman filter is still appropriate. Initially, we have $b$ is independent of $z_0$. The model therefore is:

at $t=0$

$$\mathrm{Var}\begin{vmatrix} b_0 \\ z_0 \end{vmatrix} = k\begin{vmatrix} I_L & 0 \\ 0 & V_1(0|0) \end{vmatrix} + \begin{vmatrix} 0 & 0 \\ 0 & V_0(0|0) \end{vmatrix} + O\,(k^{-1}) \tag{4.14}$$

$$\begin{vmatrix} b_{t+1} \\ z_{t+1} \end{vmatrix} = \begin{vmatrix} I_L & 0 \\ 0 & F \end{vmatrix}\begin{vmatrix} b_t \\ z_t \end{vmatrix} + \begin{vmatrix} 0 \\ G \end{vmatrix}\,\varepsilon_{t+1} \tag{4.15}$$

and $$y_t = (x_t', h_t')'\begin{vmatrix} b_t \\ z_t \end{vmatrix}, \tag{4.16}$$

where $z_0$, $V_1(0|0)$, $V_0(0|0)$, $F$, $G$ and $h_t$ are all given in Section 3.2.

## 4.2 Polynomial Algorithms

The following algorithms are used for multiplication of polynomials. These algorithms are needed in Section 4.3.

### Algorithm POLYMULT (a, b)

Consider two polynomials

$$P_1(x) = 1 - a_1 x - \dots - a_p x^p \tag{4.17}$$

and

$$P_2(x) = 1 - b_1 x - \dots - b_q x^q \tag{4.18}$$

The arguments of the algorithm are $\mathbf{a} = (a_1, \dots, a_p)'$ and $\mathbf{b} = (b_1, \dots, b_q)'$. The function POLYMULT (a,b) returns the value $c = (c_1, \dots, c_{p+q})'$, where

$$[P_1(x)][P_2(x)] = 1 - c_1 x - \dots - c_{p+q} x^{p+q}. \tag{4.19}$$

### Algorithm DPLYMLTA (a, b)

Given the input parameters as in POLYMULT, this function return a $(p+q)$ by $p$ matrix of derivatives of the result of POLYMULT $(\mathbf{a},\mathbf{b})$ with respect to $\mathbf{a}$. If $c$ is given by expansion (4.19), we have

$$\begin{aligned} \frac{\partial c_i}{\partial a_j} &= 1 \quad \text{if} \quad i=j \\ &= -b_k \quad \text{if} \quad i=j+k \quad \text{for} \quad k=1, \dots, q. \end{aligned}$$

### Algorithm POLYPOWR (a, n)

By repeated application of POLYMULT, the algorithm POLYPOWR $(\mathbf{a},n)$ computes the coefficients of $[P_1(x)]^n$, where $P_1(x)$ is given by (4.17).

### 4.3 Initialization Algorithms

The following algorithms are used to set up the initial conditions for the state-space model.

Algorithm CUE (a, b, c)

This algorithm is used to compute the components of $C_{UE}$ in (2.84). Suppose we have an ARIMA process

$$a(B)\ c(B)\theta_t = b(B)\varepsilon_t \tag{4.20}$$

where

$$a(B) = 1 - a_1 B - \dots - a_p B^p \tag{4.21}$$

$$c(B) = 1 - c_1 B - \dots - c_R B^R \tag{4.22}$$

$$b(B) = 1 - b_1\ B - \dots - b_q B^q, \tag{4.23}$$

where $c(B)$ is the differencing term, so that all the roots of $c(B)$ are on the unit circle. The $\{\varepsilon_t\}$ are independent $N(0,\sigma^2)$.

The function $CUE(\mathbf{a},\mathbf{b},\mathbf{c})$ returns values $d_1, \dots, d_r$ where $r = \max(p+R-1,\ q+1)$ and

$$\sigma^2\ d_i = \mathrm{Cov}(u_t,\ \varepsilon_{t-i+1}), \tag{4.24}$$

where $u_t = c(B)\theta_t$.

The $d_i$'s are derived by multiplying

$$a(B)\ u_t = b(B)\ \varepsilon_t \tag{4.25}$$

by $\varepsilon_{t-i+1}$ and taking expectations. The computations are

$$d_1 = 1 \tag{4.26}$$

$$d_i = a_1\ d_{i-1} + a_2\ d_{i-2} + \dots + a_{i-1}d_1 - b_{i-1} \quad \text{for} \quad i=2, \dots, r, \tag{4.27}$$

where $b_{q+1} = \dots = b_{r-1} = 0$ if $r > q+1$.

Algorithms DCUEA (**a**,**b**,**c**) and DCUEB (**a**,**b**,**c**)

The functions DCUEA (**a**,**b**,**c**) and DCUEB (**a**,**b**,**c**) compute the derivatives of CUE (**a**,**b**,**c**) with respect to **a** and **b**, respectively. The results have $r$ rows, and $p$ columns and $q$ columns, respectively, where $r = \max(p+R-1,\ q+1)$. From (4.26) and (4.27) we have

$$\frac{\partial d_i}{\partial a_j} = a_1 \frac{\partial d_{i-1}}{\partial a_j} + \dots + a_{i-1} \frac{\partial d_1}{\partial a_j} + d_{i-j}, \quad \text{for } i>1 \text{ and } j<i; \tag{4.28}$$

$$\frac{\partial d_i}{\partial b_j} = a_1 \frac{\partial d_{i-1}}{\partial b_j} + \dots + a_{i-1} \frac{\partial d_1}{\partial b_j}, \quad \text{for } i>1 \text{ and } j<i-1; \tag{4.29}$$

$$\frac{\partial d_i}{\partial b_{i-1}} = a_1 \frac{d_{i-1}}{\partial b_{i-1}} + \dots + a_{i-1} \frac{\partial d_1}{\partial b_{i-1}} - 1, \text{ for } i>1; \tag{4.30}$$

$$\frac{\partial d_i}{\partial b_j} = 0 \qquad \text{otherwise.} \tag{4.31}$$

Algorithm VU (**a**,**b**)

Using the model (4.25), the function VU(**a**,**b**) computes the vector $v_1, v_2, \dots, v_{p+1}$ where

$$\sigma^2 v_i = \mathrm{Cov}(u_t, u_{t-i+1}).$$

Multiplying expression (4.25) by $u_{t-i+1}$ and taking expectations yields the following system of equations.

$$\begin{aligned}
v_1 - a_1 v_2 - a_2 v_3 - \dots - a_p v_{p+1} &= d_1 - b_1 d_2 - \dots - b_{r-1} d_r \\
v_2 - a_1 v_1 - a_2 v_2 - \dots - a_p v_p &= - b_1 d_1 - \dots - b_{r-1} d_{r-1} \\
v_3 - a_1 v_2 - a_2 v_1 - \dots - a_p v_{p-1} &= - b_2 d_1 - \dots - b_{r-1} d_{r-2} \\
&\vdots \\
v_{p+1} - a_1 v_p - a_2 v_{p-1} - \dots - a_p v_1 &= - b_p d_1 - \dots - b_{r-1} d_{r-p},
\end{aligned} \tag{4.33}$$

where $b_{q+1} = \ldots = b_{r-1} = 0$ if $r>q+1$, and $d_1, \ldots, d_r$ is the result of CUE.

Algorithms DVUA (**a**,**b**) and DVUB (**a**,**b**)

The functions DVUA (**a**,**b**) and DVUB (**a**,**b**) compute the derivatives of VU(**a**,**b**) with respect to **a** and **b** respectively. The results have p+1 rows, and p columns and q columns, respectively. The system of equations (4.33) may be written as

$$Dv = e \tag{4.34}$$

where D is the (p+1) by (p+1) matrix on the left hand side of (4.33) and e is the (p+1)-dimensional column vector on the right hand side. We therefore have

$$\frac{\partial v}{\partial a_i} = D^{-1}\left|\frac{\partial e}{\partial a_i} - \left(\frac{\partial D}{\partial a_i}\right) v\right| \tag{4.35}$$

and

$$\frac{\partial v}{\partial b_i} = D^{-1}\left(\frac{\partial e}{\partial b_i}\right). \tag{4.36}$$

To compute $\partial e/\partial a_i$ and $\partial e/\partial b_i$ we need the results of DCUEA and DCUEB. Note that $\partial D/\partial a_i$ does not depend on the value of **a**, only on its dimensionality.

We have, for $i>1$,

$$\frac{\partial e_i}{\partial a_j} = - b_{i-1} \frac{\partial d_1}{\partial a_j} - \ldots - b_{r-1} \frac{\partial d_{r-i+1}}{\partial a_j}, \tag{4.37}$$

$$\frac{\partial e_i}{\partial b_j} = - b_{i-1} \frac{\partial d_1}{\partial b_j} - \ldots - b_{r-1} \frac{\partial d_{r-i+1}}{\partial b_j} - d_{j-i+2}. \tag{4.38}$$

The expressions for $\partial e_1/\partial a_j$ and $\partial e_1/\partial b_j$ can be derived analogously.

Algorithm INITV **(a,b,c)**

For model (4.20), the function INITV **(a,b,c)** returns a $2r$ by $r$ matrix where $r = \max(p+R,\ q+1)$. The first $r$ rows of the result correspond to $\mathbf{V}_1(0|0)$ in (2.83), and the last $r$ rows correspond to $\mathbf{V}_0(0|0)$ in (2.84) for $\sigma^2=1$.

The algorithm proceeds by constructing the matrix $\mathbf{M}$ from (2.78) and $\mathbf{A}_1$ and $\mathbf{A}_2$ from (2.73) and (2.74). Note that $\mathbf{A}_1$ contains $a_2^\star, \ldots, a_{p+R}^\star$ which is obtained from POLYMULT **(a,c)**. The matrix $\mathbf{V}_u$ in (2.84) is given by

$$\mathbf{V}_u = \begin{vmatrix} v_1 & v_2 & \cdots & v_{\max(p,1)} \\ v_2 & v_1 & & v_{p-1} \\ \vdots & \vdots & & \vdots \\ v_{\max(p,1)} & v_{p-1} & & v_1 \end{vmatrix}, \tag{4.39}$$

where $\{v_i\}$ is the solution to (4.34).

The matrix $C_{UE}$ in (2.84) is given by

$$C_{UE} = \begin{vmatrix} d_1 & d_2 & \cdots & d_{r-1} \\ 0 & d_1 & \cdots & d_{r-2} \\ \vdots & & & \\ 0 & 0 & \cdots & d_{r-\max(p,1)} \end{vmatrix}$$

where $\{d_i\}$ is the result of CUE**(a,b,c)**. Matrices $\mathbf{V}_1(0|0)$ and $\mathbf{V}_0(0|0)$ are then computed from (2.83) and (2.84) for $\sigma^2=1$.

Algorithms DINITVA **(a,b,c)** and DINITVB **(a,b,c)**

The results of DINITVA **(a,b,c)** and DINITVB **(a,b,c)** are a $2r$ by $r$ by $p$ array and a $2r$ by $r$ by $q$ array, respectively, corresponding to the derivatives of the result of INITV **(a,b,c)** with respect to **a** and **b**. Letting $\mathbf{K} = \mathbf{A}_1\mathbf{M}^{-1}$, we have

$$\frac{\partial K}{\partial \mathbf{a}} = \left| \left(\frac{\partial A_1}{\partial \mathbf{a}^*}\right)\left(\frac{\partial \mathbf{a}^*}{\partial \mathbf{a}}\right) \right| M^{-1}, \tag{4.41}$$

where $\partial\overset{*}{\mathbf{a}}/\partial\mathbf{a}$ is the result of DPLYMLTA $(\mathbf{a},\mathbf{c})$. From (4.41) we obtain $\partial K_1/\partial a_i$ and $\partial K_2/\partial a_i$, being the first $\max(p,1)$ columns and the last R columns of $\partial K/\partial a_i$, respectively. The expressions for the derivatives of $V_u$ and CUE can be obtained from DVUA$(\mathbf{a},\mathbf{b})$, DVUB$(\mathbf{a},\mathbf{b})$, DCUEA$(\mathbf{a},\mathbf{b},\mathbf{c})$ and DCUEB$(\mathbf{a},\mathbf{b},\mathbf{c})$ applied to (4.39) and (4.40).

Thus, we have

$$\frac{\partial V_1(0|0)}{\partial a_i} = \left(\frac{\partial K_2}{\partial a_i}\right) K_2' + K_2\left(\frac{\partial K_2}{\partial a_i}\right)', \tag{4.42}$$

$$\frac{\partial V_1(0|0)}{\partial b_i} = 0, \tag{4.43}$$

$$\begin{aligned}\frac{\partial V_0(0|0)}{\partial a_i} &= \left(\frac{\partial K_1}{\partial a_i}\right) V_u K_1' + K_1 \frac{\partial V_u}{\partial a_i} K_1' + K_1 V_u \left(\frac{\partial K_1}{\partial a_i}\right)' \\ &+ \left(\frac{\partial K_1}{\partial a_i}\right) C_{UE}\, A_2' + K_1\left(\frac{\partial C_{UE}}{\partial a_i}\right) A_2' \\ &+ A_2\left(\frac{\partial C_{UE}}{\partial a_i}\right)' K_1' + A_2\, C_{UE}' \left(\frac{\partial K_1}{\partial a_i}\right)',\end{aligned} \tag{4.44}$$

$$\frac{\partial V_0(0|0)}{\partial b_i} = K_1\left(\frac{\partial V_u}{\partial b_i}\right) K_1' + K_1\left(\frac{\partial C_{UE}}{\partial b_i}\right) A_2' + K_1 C_{UE}\left(\frac{\partial A_2}{\partial b_i}\right)'$$

$$+ \left(\frac{\partial A_2}{\partial b_i}\right) C_{UE}' K_1' + A_2\left(\frac{\partial C_{UE}}{\partial b_i}\right)' K_1'$$

$$+ \left(\frac{\partial A_2}{\partial b_i}\right) A_2' + A_2\left(\frac{\partial A_2}{\partial b_i}\right)'. \qquad (4.45)$$

Algorithm SETUP $(\lambda, \alpha, \nu, \beta, \phi, \psi, \sigma^2, \tau^2, s, d, D, L)$

Consider now the model given by (4.1) to (4.10), including the regression coefficients, **b**. SETUP returns a $(2r+1)$ by $r$ matrix, where

$$r = \max(sP + p + sD + d,\ sQ + q+1) + \max(m,\ n+1) + L \qquad (4.46)$$

The first row of the result is $\mathbf{m}_0(0|0) = \mathbf{0}$.

The next $r$ rows correspond to $V_1(0|0)$ for the $r$-dimensional state vector and the last $r$-rows correspond to $V_0(0|0)$.

The state vector is made up of three parts. The first $L$ components correspond to the regression coefficients, **b**. The next $\max(sP + p + sD + d,$ $sQ + q+1)$ components correspond to the ARIMA model for $\{\theta_t\}$ given by (4.1) and the last $\max(m, n+1)$ components correspond to the ARMA model for the $\{e_t\}$ given by (4.2).

The algorithm SETUP proceeds as follows.

1. Let $\lambda^* = (\lambda_1^*, \ldots, \lambda_{sP}^*)$ be defined as

$$\lambda_s^* = \lambda_1$$

$$\lambda_{2s}^* = \lambda_2$$

$$\vdots$$

$$\lambda_{Ps}^* = \lambda_P$$

$$\lambda_i^* = 0 \qquad \text{otherwise.}$$

Similarly, let $\nu^* = (\nu_1^*, \ldots, \nu_{sQ}^*)$ be defined as

$$\nu_s^* = \nu_1$$

$$\vdots$$

$$\nu_{Qs}^* = \nu_Q$$

$$\nu_i^* = 0 \qquad \text{otherwise.}$$

Let $\nabla_s^* = (0, 0, \ldots, 0, 1)$ an $s$-dimensional vector.

2. Let **a** be the result of POLYMULT $(\lambda^*, \alpha)$; let **b** be the result of POLYMULT $(\nu^*, \beta)$; let **c** be the result of

$$\text{POLYMULT (POLYPOWR}(\nabla_s^*, D)\text{, POLYPOWR }(1,d)\text{.}$$

3. Compute INITV (**a**, **b**, **c**). Let the result be

$$\begin{vmatrix} V_{1,1} \\ V_{0,1} \end{vmatrix}.$$

Compute INITV $(\theta, \Psi,$ null vector). Let the result be

$$\begin{vmatrix} 0 \\ V_{0,2} \end{vmatrix} .$$

4. The $(2r + 1)$ by $r$ matrix result of SETUP is:

$$\begin{vmatrix} 0 & 0 & 0 \\ I_L & 0 & 0 \\ 0 & V_{1,1} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \sigma^2 V_{0,1} & 0 \\ 0 & 0 & \tau^2 V_{0,2} \end{vmatrix} . \qquad (4.47)$$

<u>Algorithm DSETUP ($\lambda$, $\alpha$, $\nu$, $\beta$, $\phi$, $\psi$, $\sigma^2$, $\tau^2$, s, d, D, L)</u>

Algorithm DSETUP computes the derivatives of the result of SETUP with respect to the unknown parameters, $\lambda$, $\alpha$, $\nu$, $\beta$, and $\sigma^2$. The result is a $(2r+1)$ by $r$ by $(P + p + Q + q + 1)$ array. To compute this, we need: $\partial \mathbf{a}/\partial \alpha$ given by DPLYMLTA ($\alpha$, $\lambda^*$), $\partial \mathbf{a}/\partial \lambda$ given by every s-th column of DPLYMLTA ($\lambda^*$, $\alpha$), $\partial \mathbf{b}/\partial \beta$ given by DPLYMLTA ($\beta$, $\nu^*$) and $\partial \mathbf{b}/\partial \nu$ given by every s-th column of DPLYMLTA ($\nu^*$, $\beta$).

Computing the results of DINITVA $(\mathbf{a},\mathbf{b},\mathbf{c})$ and DINITVB $(\mathbf{a},\mathbf{b},\mathbf{c})$ we obtain

$$\frac{\partial V_{1,1}}{\partial \lambda} = \left(\frac{\partial V_{1,1}}{\partial \mathbf{a}}\right)\left(\frac{\partial \mathbf{a}}{\partial \lambda}\right)$$

$$\frac{\partial V_{1,1}}{\partial \alpha} = \left(\frac{\partial V_{1,1}}{\partial \mathbf{a}}\right)\left(\frac{\partial \mathbf{a}}{\partial \alpha}\right)$$

$$\frac{\partial(\sigma^2 V_{0,1})}{\partial \lambda} = \sigma^2\left(\frac{\partial V_{0,1}}{\partial \mathbf{a}}\right)\left(\frac{\partial \mathbf{a}}{\partial \lambda}\right)$$

$$\frac{\partial(\sigma^2 V_{0,1})}{\partial \alpha} = \sigma^2\left(\frac{\partial V_{0,1}}{\partial \mathbf{a}}\right)\left(\frac{\partial \alpha}{\partial \alpha}\right)$$

$$\frac{\partial(\sigma^2 V_{0,1})}{\partial \nu} = \sigma^2\left(\frac{\partial V_{0,1}}{\partial \mathbf{b}}\right)\left(\frac{\partial \mathbf{b}}{\partial \nu}\right)$$

$$\frac{\partial(\sigma^2 V_{0,1})}{\partial\beta} = \sigma^2\left(\frac{\partial V_{0,1}}{\partial b}\right)\left(\frac{\partial b}{\partial\beta}\right)$$

$$\frac{\partial(\sigma^2 V_{0,1})}{\partial\sigma^2} = V_{0,1}$$

All other derivatives of (4.47) will respect to the unknown parameters are zero.

## 4.4 Likelihood Function Algorithms

The algorithm to compute $m(0|0)$ and $V(0|0)$ and their derivatives is given by SETUP and DSETUP in Section 4.3. The model is completed as follows. We let $F$ be the $r$ by $r$ matrix

$$F = \begin{vmatrix} I_L & 0 & 0 \\ 0 & F_1 & 0 \\ 0 & 0 & F_2 \end{vmatrix}, \tag{4.47}$$

where $r$ is given (4.46), $F_1$ and $F_2$ are given by (3.9) and (3.10). We let $G$ be the $r$ by 2 matrix

$$G = \begin{vmatrix} 0 & 0 \\ G_1 & 0 \\ 0 & G_2 \end{vmatrix}, \tag{4.48}$$

where $G_1 = (1, -b_1, \ldots, -b_{q+sQ})'$ for $b$ being the result of POLYMULT($\nu^*$,$\beta$), and $G_2 = (1, -\psi_1, \ldots, \psi_n)'$. Finally, we let

$$h_t' = (x_t', 1, \ldots, 0, k_t^{-1}, 0, \ldots, 0). \tag{4.49}$$

The modified Kalman filter recursions were given in section (2.3) so we will not give details here.

The derivatives will also be required. In terms of storage requirements, it is only necessary to keep the most recent version of $m$ and $V$. The $\{a_t\}$ and $\{v_0(t)\}$ and its derivatives will be needed for all $t$ such that $v_1(t) = 0$.

To facilitate the process, it is worthwhile to have algorithms FMULT(x) and DFMULTA(x) which compute the results of $Fx$, $(\partial Fx)/\partial\lambda$ and $(\partial Fx)/\partial\alpha$, where $x$ is an r-dimensional vector.

Finally, algorithms should be set up to compute the marginal log-likelihood functioning (2.61), its derivatives (2.62) and the information matrix (2.64).

### 4.5 Other Algorithms

The details of the Davidon-Fletcher-Powell algorithm are given in Section 3.5. Computations for fixed point smoothing are given in Section 3.3 and confidence intervals are computed as in Section 3.4. It should be noted that the addition of the regression coefficients to the model does not change the general discussion of those sections.

## 5. FURTHER RESEARCH

In this paper we have given a detailed discussion of methods to incorporate survey errors in ARIMA modelling. Other models which can be formulated within the state-space framework could use a similar approach.

A suggestion was given in Section 3.5 for maximizing the likelihood function but research into alternatives would be useful. Also the confidence intervals in Section 3.4 used asymptotic approximations whose validity could be checked by simulations for finite samples.

It was suggested in this paper that the survey errors can often be approximated by an ARMA process, at least up to a multiplicative constant. Methods for estimating these parameters from various survey designs have not been well developed. Also the confidence intervals have ignored the variation due to the estimation of the survey error variances. This topic deserves further study.

## REFERENCES

Anderson, B.D.O. and J.B. Moore (1979). *Optimal Filtering.* Prentice-Hall, Englewood Cliffs, NJ.

Ansley, C.F. and R. Kohn (1985). A structured state space approach to computing the likelihood of an ARIMA process and its derivatives. *Journal of Statistical Computation and Simulation.* 21: 135-169.

Binder, D.A. and J.P. Dick (1988). Modelling and estimation for repeated surveys. *Survey Methodology Journal.* To appear.

Binder, D.A. and M.A. Hidiroglou (1988). Sampling in Time. *Handbook of Statistics, Vol. 6.* P.R. Krishnaiah and C.R. Rao, eds., Elsevier Science, Amsterdam. 187-211.

Dennis, J.E. and R.B. Schabel (1983). *Nonlinear Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice-Hall.

Gurney, M. and J.F. Daly (1965). A multivariate approach to estimation in periodic sample surveys. *Proceedings of the Social Statistics Section, American Statistical Association.* 247-257.

Harvey, A.C. and G.D.A. Phillips (1979). Maximum likelihood estimation of regression models with autoregressive-moving average disturbances. *Biometrika.* 66: 49-58.

Jones, R.G. (1980). Best linear unbiased estimators for repeated surveys. *Journal of the Royal Statistical Society, Series B.* 42: 221-226.

Kohn, R. and C.F. Ansley (1986). Estimation, prediction and interpolation for ARIMA models with missing data. *Journal of the American Statistical Association.* 81: 751-761.