

# Fast estimation methods for time series models in state-space form

Alfredo Garcia-Hiernaux\*

José Casals

Miguel Jerez

March 28, 2007

## Abstract

We propose two new, fast and stable methods to estimate time series models written in their equivalent state-space form. They are useful both, to obtain adequate initial conditions for a maximum-likelihood iteration, or to provide final estimates when maximum-likelihood is considered inadequate or computationally expensive. The state-space foundation of these procedures provides flexibility, as they can be applied to any linear fixed-coefficients model, such as ARIMA, VARMAX or structural time series models. A simulation exercise shows that their computational costs and finite-sample performance are very good.

**Keywords:** State-space models, subspace methods, Kalman filter, system identification, maximum-likelihood

---

\*Corresponding author. Departamento de Fundamentos del Análisis Económico II. Facultad de Ciencias Económicas. Campus de Somosaguas. 28223 Madrid (SPAIN). email: agarciah@ccee.ucm.es, tel: (+34) 91 394 26 26, fax: (+34) 91 394 25 91. The authors gratefully acknowledge financial support from Ministerio de Educación y Ciencia, ref. SEJ2005-07388.

# 1 Introduction

Many model estimation methods are based either on Least Squares (LS) or Maximum Likelihood (ML). The LS approach has clear advantages in terms of computational simplicity, stability and cost. However it is limited in scope, as it cannot be directly applied to many models such as those with moving average terms or multiplicative seasonality. On the other hand ML typically requires iterative optimization, so its statistical efficiency and wide scope is somewhat balanced with increased instability, complexity and computational cost. Specifically when modeling high-frequency time series, such as those generated by financial markets, it may be hard to accept the assumptions and costs implied by gaussian ML.

As the pros and cons of LS and ML are complementary, many works focus on devising methods that provide the best of both worlds. For example, Spliid (1983), Koreisha and Pukkila (1989, 1990), Flores and Serrano (2002) and Dufour and Pelletier (2004) extend ordinary and generalized LS to VARMAX modeling. Also Lütkepohl and Poskitt (1996) propose a LS-based method to specify and estimate a canonical VARMA structure. Finally, Francq and Zakoïan (2000) provide weak GARCH representations that can be estimated using two-stage LS. All these methods share two substantial limitations. First, they avoid iteration when there are no exclusion constraints, but imposing such restrictions requires iterating and this partially offsets their computational advantage. Second, these procedures are typically devised for specific parameterizations, such as VARMAX, and their imple-

mentation in other frameworks is not trivial.

In this paper we propose two fast and stable algorithms to estimate time series models written in their equivalent State-Space (SS) form. They are useful, either to obtain adequate initial conditions for ML iteration, or to provide final estimates when ML is considered inadequate or computationally expensive. On the other hand, their SS foundations provide a wide scope, as they can be applied to any linear fixed-coefficients model with an equivalent SS form.

Both algorithms are based on subspace methods, see Favoreel et al. (2000), Katayama (2005), or Bauer (2005). The first method, SUBEST1, consists of estimating the parametric matrices in the SS model by solving a weighted reduced-rank LS problem, defined over a set of subspace regressions. Iteration is required when there are constraints relating the parameters in the standard and SS representation of the model. Any exclusion or fixed-value constraint is treated in the same way. Montecarlo experiments show that this method is very fast and provides estimates close to those of ML in finite samples. However, when compared with ML, it tends to underestimate the MA parameters.

The second and more sophisticated procedure, SUBEST2, provides a gaussian ML solution to the subspace regressions. Simulations show that it improves over SUBEST1, both in precision and dispersion of the estimates, while keeping a substantial advantage over ML in terms of computational

cost.

The structure of the paper is as follows. Section 2 defines the basic notation and previous results. Building on them, Section 3 derives the estimation algorithms and Section 4 discusses their performance in finite samples. In Section 5 we discuss the conditions under which each method is more adequate and indicate how to obtain a free MATLAB Toolbox implementing them.

## 2 Notation and previous results

### 2.1 General state-space model

Consider an  $m$ -vector of endogenous outputs,  $\mathbf{z}_t$ , which is related to its past and to the current and past values of an  $r$ -vector of exogenous inputs,  $\mathbf{u}_t$ , through a time series model depending on a vector of unknown constant parameters,  $\boldsymbol{\beta}$ . Assume also that this model can be written in an equivalent SS representation, such as:

$$\mathbf{x}_{t+1} = \boldsymbol{\Phi}\mathbf{x}_t + \boldsymbol{\Gamma}\mathbf{u}_t + \mathbf{E}\mathbf{w}_t \quad (1a)$$

$$\mathbf{z}_t = \mathbf{H}\mathbf{x}_t + \mathbf{D}\mathbf{u}_t + \mathbf{C}\mathbf{v}_t \quad (1b)$$

where the matrices  $\boldsymbol{\Phi}$ ,  $\boldsymbol{\Gamma}$ ,  $\mathbf{E}$ ,  $\mathbf{H}$ ,  $\mathbf{D}$  and  $\mathbf{C}$  depend on  $\boldsymbol{\beta}$ ;  $\mathbf{x}_t \in \mathbb{R}^n$  is the state vector and,  $\mathbf{w}_t \in \mathbb{R}^n$  and  $\mathbf{v}_t \in \mathbb{R}^m$  are uncorrelated sequences of gaussian errors, independent of  $\mathbf{u}_t$ , with zero mean and such that:  $E(\mathbf{w}_t\mathbf{w}_t') = \mathbf{Q}$ ,  $E(\mathbf{v}_t\mathbf{v}_t') = \mathbf{R}$ ,  $E(\mathbf{v}_t\mathbf{w}_t') = \mathbf{S}$ . We will also assume this model to be minimal

and stable. Minimality means that there is no SS model that can realize  $\mathbf{z}_t$  with less than  $n$  states. Stability requires all the eigenvalues of  $\Phi$  to lie inside the unit circle.

Subspace methods derive from a matrix representation of the system (1a-1b). To obtain it, note that a recursive substitution in (1a) yields,

$$\mathbf{x}_t = \Phi^t \mathbf{x}_0 + \sum_{j=0}^{t-1} \Phi^j \mathbf{E} \mathbf{w}_{t-j-1} + \sum_{j=0}^{t-1} \Phi^j \Gamma \mathbf{u}_{t-j-1} \quad (2)$$

and substituting (2) into the observation equation (1b), we get:

$$\mathbf{z}_t = \mathbf{H} \Phi^t \mathbf{x}_0 + \mathbf{H} \sum_{j=0}^{t-1} \Phi^j \Gamma \mathbf{u}_{t-j-1} + \mathbf{D} \mathbf{u}_t + \mathbf{H} \sum_{j=0}^{t-1} \Phi^j \mathbf{E} \mathbf{w}_{t-j-1} + \mathbf{C} \mathbf{v}_t \quad (3)$$

so the endogenous variable,  $\mathbf{z}_t$ , depends on the initial state vector,  $\mathbf{x}_0$ , the present and past values of the inputs,  $\mathbf{u}_t$ , and both errors  $\mathbf{w}_t$  and  $\mathbf{v}_t$ . Under these conditions, equation (3) can be written in matrix form as,

$$\mathbf{Z}_f = \mathbf{O}_i \mathbf{X}_f + \mathbf{H}_i^u \mathbf{U}_f + \mathbf{H}_i^w \mathbf{W}_f + \mathbf{H}_i^v \mathbf{V}_f \quad (4)$$

The regression model (4) includes both, data-dependent and parameter-dependent matrices. The data-dependent matrices are the Block-Hankel matrices (BHMs)  $\mathbf{Z}_f$ ,  $\mathbf{U}_f$ ,  $\mathbf{W}_f$  and  $\mathbf{V}_f$ , as well as the state sequence  $\mathbf{X}_f$ . The row space of these matrices, which is denoted by their subscript and will be defined in (6), depends on the parameter  $i$ . In this work, we will determine  $i$  as the nearest integer to  $\log T$ , being  $T$  the sample size. This heuristic

choice is supported by empirical experience, but our procedures could be implemented with any other choice, such as those suggested by Van Overschee and De Moor (1994), Deistler et al. (1995), Ljung (1999) or Bauer (2005).

The BHM  $\mathbf{Z}_f$  organizes the “future” values of the outputs from  $t = i$  onwards. Its structure is:

$$\mathbf{Z}_f = \mathbf{Z}_{i:2i-1} = \begin{pmatrix} z_i & z_{i+1} & \cdots & z_{T-i} \\ z_{i+1} & z_{i+2} & \cdots & z_{T-i+1} \\ \vdots & \vdots & & \vdots \\ z_{2i-1} & z_{2i} & \cdots & z_{T-1} \end{pmatrix} \quad (5)$$

and the complement of (5) is the matrix of “past” values  $\mathbf{Z}_p$ , which structure is:

$$\mathbf{Z}_p = \mathbf{Z}_{0:i-1} = \begin{pmatrix} z_0 & z_1 & \cdots & z_{T-2i} \\ z_1 & z_2 & \cdots & z_{T-2i+1} \\ \vdots & \vdots & & \vdots \\ z_{i-1} & z_i & \cdots & z_{T-i-1} \end{pmatrix} \quad (6)$$

The matrix  $\mathbf{Z}_p$  will be employed in Section 3. The other BHMs  $\mathbf{U}_f$ ,  $\mathbf{W}_f$  and  $\mathbf{V}_f$  have the same structure as  $\mathbf{Z}_f$  but depending, respectively, on  $\mathbf{u}_t$ ,  $\mathbf{w}_t$  and  $\mathbf{v}_t$  instead of  $z_t$ . For short hand notation we denote, any BHM  $\mathbf{A}$  as:  $\mathbf{A} = \mathbf{A}_{0:2i-1}$ ,  $\mathbf{A}_{p+} = \mathbf{A}_{0:i}$ ,  $\mathbf{A}_{pr} = \mathbf{A}_{i:i}$  and  $\mathbf{A}_{f+} = \mathbf{A}_{i+1:2i-1}$ .

The last data-dependent matrix is the state sequence  $\mathbf{X}_f$ , given by:

$$\mathbf{X}_f = (\mathbf{x}_i \quad \mathbf{x}_{i+1} \quad \mathbf{x}_{i+2} \quad \cdots \quad \mathbf{x}_{T-i}), \quad \text{with } f = i. \quad (7)$$

The states in  $\mathbf{X}_f$  are unobserved, so they must be estimated from the sample. In Section 3.1 we describe how to do this.

On the other hand, equation (4) also depends on the following parameter-related matrices: the Extended Observability matrix,  $\mathbf{O}_i$ , the lower-block diagonal Toeplitz matrices,  $\mathbf{H}_i^u$  and  $\mathbf{H}_i^w$ , and the block-diagonal matrix  $\mathbf{H}_i^v$ . These matrices are defined by:

$$\mathbf{O}_i = \left( \mathbf{H}' \quad (\mathbf{H}\Phi)' \quad (\mathbf{H}\Phi^2)' \quad \dots \quad (\mathbf{H}\Phi^{i-1})' \right)' \in \mathbb{R}^{im \times n}, \quad (8)$$

$$\mathbf{H}_i^u = \begin{pmatrix} \mathbf{D} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{H}\Gamma & \mathbf{D} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{H}\Phi\Gamma & \mathbf{H}\Gamma & \mathbf{D} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}\Phi^{i-2}\Gamma & \mathbf{H}\Phi^{i-3}\Gamma & \mathbf{H}\Phi^{i-4}\Gamma & \dots & \mathbf{D} \end{pmatrix} \in \mathbb{R}^{im \times ir}, \quad (9)$$

$$\mathbf{H}_i^w = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{H}\mathbf{E} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{H}\Phi\mathbf{E} & \mathbf{H}\mathbf{E} & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{H}\Phi^{i-2}\mathbf{E} & \mathbf{H}\Phi^{i-3}\mathbf{E} & \mathbf{H}\Phi^{i-4}\mathbf{E} & \dots & \mathbf{0} \end{pmatrix} \in \mathbb{R}^{im \times im}, \quad (10)$$

and

$$\mathbf{H}_i^v = \begin{pmatrix} \mathbf{C} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{C} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{C} \end{pmatrix} \in \mathbb{R}^{im \times im} \quad (11)$$

## 2.2 Steady-state innovations model

Our algorithms become simpler and more efficient when applied to a special SS representation known as steady-state innovations model, hereafter “innovations model”. An innovations model is given by:

$$\mathbf{x}_{t+1} = \mathbf{\Phi} \mathbf{x}_t + \mathbf{\Gamma} \mathbf{u}_t + \mathbf{E} \psi_t \quad (12a)$$

$$\mathbf{z}_t = \mathbf{H} \mathbf{x}_t + \mathbf{D} \mathbf{u}_t + \psi_t \quad (12b)$$

Note that this model can be viewed as a special case of (1a-1b) with  $\mathbf{C} = \mathbf{I}_m$ , which is an identity  $m$ -by- $m$  matrix, and  $\mathbf{w}_t = \mathbf{v}_t = \psi_t$ . Besides the assumptions made over model (1a-1b), we add a strict minimum-phase condition, so all the eigenvalues ( $\mathbf{\Phi} - \mathbf{E}\mathbf{H}$ ) lie inside the unit circle.

The innovations form (12a-12b) is convenient since it is general and it is computationally efficient.

It is general because of two reasons. First, many time series models, including rational transfer functions and VARMAX, can be directly written in the form (12a-12b), see Hannan and Deistler (1988), since they present only

one source of unobserved errors. On the other hand, any model in the form (1a-1b) such as, e.g., the Structural Time Series Models (STSM, Harvey, 1989), can be written under weak assumptions in its equivalent innovations form (see Casals et al., 1999, Theorem 1).

It is efficient because, its structure presents less parameters than the general form (1a-1b), reducing the complexity, the computational cost and improving the likelihood conditioning. Specifically, when the model is written in the innovations form (12a-12b), equation (4) collapses to the simpler form:

$$\mathbf{Z}_f = \mathbf{O}_i \mathbf{X}_f + \mathbf{H}_i^u \mathbf{U}_f + \mathbf{H}_i^\psi \mathbf{\Psi}_f \quad (13)$$

where  $\mathbf{H}_i^w$  and  $\mathbf{H}_i^v$  are replaced with  $\mathbf{H}_i^\psi$ , that is just as (10) but with  $\mathbf{I}_m$  in the main block diagonal.

### 3 Main results

The basic problem consists of estimating  $\boldsymbol{\beta}$ , which is the vector of parameters in the original model. The parametric matrices  $\boldsymbol{\Phi}$ ,  $\boldsymbol{\Gamma}$ ,  $\mathbf{E}$ ,  $\mathbf{H}$ ,  $\mathbf{D}$ ,  $\mathbf{C}$ ,  $\mathbf{Q}$ ,  $\mathbf{R}$  and  $\mathbf{S}$  in the SS representation (1a-1b) of the model will be, in general, nonlinear functions of  $\boldsymbol{\beta}$ . In this section we describe two new algorithms, based on subspace methods, to estimate all these parametric matrices by iterating on  $\boldsymbol{\beta}$ . To this end, we will start by discussing how can one compute the state sequences in the matrix (7).

### 3.1 Estimation of the state sequences

As the final result does not depend on the specific SS representation employed, we will develop it from the simpler regression model (13). It implies that the expectation of future output values, conditional to  $\mathbf{U}$ , which contains the past and future information of the observed inputs, and to  $\mathbf{Z}_p$ , which contains the past information of the output, is,

$$\mathbb{E}(\mathbf{Z}_f|\mathbf{U}, \mathbf{Z}_p) = \mathbf{O}_i\mathbb{E}(\mathbf{X}_f|\mathbf{U}, \mathbf{Z}_p) + \mathbf{H}_i^u\mathbb{E}(\mathbf{U}_f|\mathbf{U}, \mathbf{Z}_p) + \mathbf{H}_i^\psi\mathbb{E}(\boldsymbol{\Psi}_f|\mathbf{U}, \mathbf{Z}_p) \quad (14)$$

where  $\mathbf{U}_f$  is included in  $\mathbf{U}$ . As we assumed in Section 2, the innovations  $\boldsymbol{\psi}_t$  are independent of the inputs and, by construction,  $\mathbb{E}(\mathbf{z}_{t-j}\boldsymbol{\psi}_t) = \mathbf{0} \forall j > 0$ , so (14) simplifies to,

$$\mathbb{E}(\mathbf{Z}_f|\mathbf{U}, \mathbf{Z}_p) = \mathbf{O}_i\mathbb{E}(\mathbf{X}_f|\mathbf{U}, \mathbf{Z}_p) + \mathbf{H}_i^u\mathbf{U}_f \quad (15)$$

Rearranging the terms of this equation, we obtain,

$$\mathbb{E}(\mathbf{X}_f|\mathbf{U}, \mathbf{Z}_p) = \mathbf{O}_i^\dagger[\mathbb{E}(\mathbf{Z}_f|\mathbf{U}, \mathbf{Z}_p) - \mathbf{H}_i^u\mathbf{U}_f] \quad (16)$$

which is an expression for the states conditional to the past of the output and to the observed inputs. As all these variables are observable, from (16) we can estimate the states by means of the orthogonal projections:

$$\hat{\mathbf{X}}_f = \mathbf{O}_i^\dagger(\mathbf{Z}_f\Pi_{\mathbf{U}, \mathbf{Z}_p} - \mathbf{H}_i^u\mathbf{U}_f) \quad (17)$$

where  $\mathbf{O}_i^\dagger$  is the Moore-Penrose pseudo-inverse of  $\mathbf{O}_i$ . Expression (17) contains parameters in  $\mathbf{O}_i^\dagger$  and  $\mathbf{H}_i^u$ , while  $\mathbf{Z}_f \mathbf{\Pi}_{U, \mathbf{Z}_p}$  denotes the orthogonal projection of the stacked matrix  $\begin{pmatrix} U \\ \mathbf{Z}_p \end{pmatrix}$ .

## 3.2 Algorithm SUBEST1

### 3.2.1 Models in innovations form

We will now set a generalized non-linear least square problem for all the parameters  $\beta$ . To do so, we start by writing equation (13) as:

$$\begin{aligned} \mathbf{Z}_{f+} &= \mathbf{O}_{i-1} [(\Phi - \mathbf{E}H)\mathbf{X}_f + \mathbf{E}\mathbf{Z}_{pr} + (\Gamma - \mathbf{E}D)\mathbf{U}_{pr}] \\ &+ \mathbf{H}_{i-1}^u \mathbf{U}_{f+} + \mathbf{H}_{i-1}^\psi \Psi_{f+} \end{aligned} \quad (18)$$

see Appendix A, where  $\mathbf{X}_f$  and  $\Psi_{f+}$  are unobserved components. The first addend in the square brackets corresponds to the states related to the past,  $\mathbf{Z}_p$  and  $\mathbf{U}_p$  (see equation A.3). The second and third addends are updating terms related to the present. Now, the expectation of (18) conditional to  $\mathbf{Z}_{p+}$  is:

$$\begin{aligned} \mathbb{E}(\mathbf{Z}_{f+} | \mathbf{U}, \mathbf{Z}_{p+}) &= \mathbf{O}_{i-1} [(\Phi - \mathbf{E}H)\mathbb{E}(\mathbf{X}_f | \mathbf{U}, \mathbf{Z}_{p+}) \\ &+ \mathbf{E}\mathbb{E}(\mathbf{Z}_{pr} | \mathbf{U}, \mathbf{Z}_{p+}) + (\Gamma - \mathbf{E}D)\mathbb{E}(\mathbf{U}_{pr} | \mathbf{U}, \mathbf{Z}_{p+})] \\ &+ \mathbf{H}_{i-1}^u \mathbb{E}(\mathbf{U}_{f+} | \mathbf{U}, \mathbf{Z}_{p+}) + \mathbf{H}_{i-1}^\psi \mathbb{E}(\Psi_{f+} | \mathbf{U}, \mathbf{Z}_{p+}) \end{aligned} \quad (19)$$

where, as  $\mathbf{X}_f$  is only related to the past, one may substitute  $\mathbb{E}(\mathbf{X}_f | \mathbf{U}, \mathbf{Z}_{p+})$  for  $\mathbb{E}(\mathbf{X}_f | \mathbf{U}, \mathbf{Z}_p)$ . Furthermore, since  $\mathbf{U}_{pr}$  and  $\mathbf{U}_{f+}$  are included in  $\mathbf{U}$ ,  $\mathbf{Z}_{pr}$

in  $\mathbf{Z}_{p+}$  and  $E(\Psi_{f+}|\mathbf{U}, \mathbf{Z}_{p+}) = \mathbf{0}$ , equation (19) simplifies to,

$$\begin{aligned} E(\mathbf{Z}_{f+}|\mathbf{U}, \mathbf{Z}_{p+}) &= \mathbf{O}_{i-1} [(\Phi - \mathbf{E}\mathbf{H})E(\mathbf{X}_f|\mathbf{U}, \mathbf{Z}_p) + \mathbf{E}\mathbf{Z}_{pr} \\ &\quad + (\Gamma - \mathbf{E}\mathbf{D})\mathbf{U}_{pr}] + \mathbf{H}_{i-1}^u \mathbf{U}_{f+} \end{aligned} \quad (20)$$

Finally, substituting  $E(\mathbf{X}_f|\mathbf{U}, \mathbf{Z}_p)$  by its estimate  $\hat{\mathbf{X}}_f$  given in (17), we obtain an estimate for the expected value of  $\mathbf{Z}_{f+}$  conditional to  $\mathbf{U}$  and  $\mathbf{Z}_{p+}$ :

$$\hat{\mathbf{Z}}_{f+} = \mathbf{O}_{i-1} [(\Phi - \mathbf{E}\mathbf{H})\hat{\mathbf{X}}_f + \mathbf{E}\mathbf{Z}_{pr} + (\Gamma - \mathbf{E}\mathbf{D})\mathbf{U}_{pr}] + \mathbf{H}_{i-1}^u \mathbf{U}_{f+} \quad (21)$$

Note that this equation includes all the parametric matrices in (12a-12b),  $\Phi$ ,  $\Gamma$ ,  $\mathbf{E}$ ,  $\mathbf{H}$  and  $\mathbf{D}$  which, therefore, can be estimated simultaneously by solving the weighted least-squares problem:

$$\min_{\{\Phi, \Gamma, \mathbf{E}, \mathbf{H}, \mathbf{D}\}} \left\| \Omega^{-\frac{1}{2}} \left( \mathbf{Z}_{f+} \Pi_{\mathbf{U}, \mathbf{Z}_{p+}} - \hat{\mathbf{Z}}_{f+} \right) \right\|_F^2 \quad (22)$$

where  $\|\cdot\|_F$  denotes de Frobenius norm and  $\Omega$  is the covariance matrix of the prediction errors, which can be written as  $\Omega = \mathbf{Z}_{f+} \Pi_{\mathbf{U}, \mathbf{Z}_{p+}}^\perp \mathbf{Z}_{f+}'$ , because  $\Pi_{\mathbf{U}, \mathbf{Z}_{p+}}^\perp$  is idempotent and where  $\Pi^\perp = \mathbf{I} - \Pi$ . Note that  $\Omega$  and  $\mathbf{Z}_{f+} \Pi_{\mathbf{U}, \mathbf{Z}_{p+}}$  are projections of data-dependent matrices, while  $\hat{\mathbf{Z}}_{f+}$ , defined in (21), depends on both, the data and the parameters. This projections can be efficiently computed using the QR decomposition (see Golub and Van Loan, 1996).

In the same way that the matrices in model (12a-12b) depend on  $\beta$ , the parametric matrices in the subspace equation (13) depend on the parameters

of the SS representation. Therefore, solving (22) requires iterative techniques such as those described in, e.g., Dennis and Schnabel (1983).

Finally, we compute the residual vector,  $\tilde{\mathbf{Z}}_{pr}$ , by shifting the time subscripts in (13):

$$\tilde{\mathbf{Z}}_{pr} = \mathbf{Z}_{pr} - \hat{\mathbf{O}}_i \hat{\mathbf{X}}_f - \hat{\mathbf{H}}_i^u \mathbf{U}_{pr} \quad (23)$$

Note that the row space dimension of  $\mathbf{Z}_{pr}$  is equal to 1, so by using just the first row of matrices  $\hat{\mathbf{O}}_i$  and  $\hat{\mathbf{H}}_i^u$ , we obtain,

$$\tilde{\mathbf{Z}}_{pr} = \mathbf{Z}_{pr} - \hat{\mathbf{H}} \hat{\mathbf{X}}_1 + \hat{\mathbf{D}} \mathbf{U}_{pr} \quad (24)$$

and, consequently,  $\hat{\mathbf{Q}} = (T - 2i + 1)^{-1} \tilde{\mathbf{Z}}_{pr} \tilde{\mathbf{Z}}'_{pr}$ , where  $T - 2i + 1$  is the number of elements of the vector  $\tilde{\mathbf{Z}}_{pr}$ , is an estimate for the innovation variance.

### 3.2.2 General SS models

SUBEST1 can be applied to SS models with more than one source of unobserved errors by using the following two-stage variant of the algorithm.

In the first step, we estimate the parameters in  $\Phi, \Gamma, \mathbf{H}$  and  $\mathbf{D}$  by solving,

$$\min_{\{\Phi, \Gamma, \mathbf{H}, \mathbf{D}\}} \left\| \Omega^{-\frac{1}{2}} \left[ \mathbf{Z}_f \Pi_{U, Z_p} - \mathbf{O}_i \hat{\mathbf{X}}_f - \mathbf{H}_i^u \mathbf{U}_f \right] \right\|_F^2 \quad (25)$$

where  $\Omega = \mathbf{Z}_f \Pi_{U, Z_p}^\perp \mathbf{Z}_f'$  and the expression for the states is given in equation (17). Note that, while in (22)  $\mathbf{E}$  is jointly estimated with the rest of parametric matrices, here it will be calculated in a second stage.

In the second step, we use the state resulting from the solution of (25) to obtain initial conditions for the Kalman filter gain,  $\mathbf{K}_i$  and the prediction error covariance,  $\mathbf{B}_i$  (see Appendix B). Then, we compute the estimates for  $\mathbf{E}$ ,  $\mathbf{C}$ ,  $\mathbf{Q}$ ,  $\mathbf{R}$  and  $\mathbf{S}$  by solving the constrained problem:

$$\min_{\{\mathbf{E}, \mathbf{C}, \mathbf{Q}, \mathbf{R}, \mathbf{S}\}} \left\| \begin{bmatrix} \underline{\mathbf{K}} \\ \underline{\mathbf{B}} \end{bmatrix} - \begin{bmatrix} \hat{\mathbf{K}}_i \\ \hat{\mathbf{B}}_i \end{bmatrix} \right\|_F^2 \quad (26)$$

where the underlined matrices were previously estimated and the matrices  $\hat{\mathbf{K}}_i$  and  $\hat{\mathbf{B}}_i$  in (26) must also satisfy the covariance equations of the Kalman filter propagated  $i$  times:

$$\begin{aligned} \hat{\mathbf{K}}_i &= (\hat{\Phi} \hat{\mathbf{P}}_i \hat{\mathbf{H}}' + \hat{\mathbf{E}} \hat{\mathbf{S}} \hat{\mathbf{C}}') \hat{\mathbf{B}}_i^{-1} \\ \hat{\mathbf{P}}_{i+1} &= \hat{\Phi} \hat{\mathbf{P}}_i \hat{\Phi}' + \hat{\mathbf{E}} \hat{\mathbf{Q}} \hat{\mathbf{E}}' - \hat{\mathbf{K}}_i \hat{\mathbf{B}}_i \hat{\mathbf{K}}_i' \\ \hat{\mathbf{B}}_i &= \hat{\mathbf{H}} \hat{\mathbf{P}}_i \hat{\mathbf{H}}' + \hat{\mathbf{C}} \hat{\mathbf{R}} \hat{\mathbf{C}}' \end{aligned} \quad (27)$$

Therefore, in this second step we compute estimates for  $\mathbf{E}$ ,  $\mathbf{C}$ ,  $\mathbf{Q}$ ,  $\mathbf{R}$  and  $\mathbf{S}$  such that they, (a) return  $\hat{\mathbf{K}}_i$  and  $\hat{\mathbf{B}}_i$  propagating equations (27)  $i$  times, and (b) are close to the values  $\underline{\mathbf{K}}$  and  $\underline{\mathbf{B}}$  resulting from the previous stage.

### 3.3 Algorithm SUBEST2

Our second algorithm requires the SS model to be written in innovations form. Actually this is not a limitation, because in each iteration we can transform the input model to the equivalent innovations form using the algorithm of Casals et al. (1999). Note that we could not do this in SUBEST1,

since the matrices related to the noise  $\mathbf{E}$ ,  $\mathbf{C}$ ,  $\mathbf{Q}$ ,  $\mathbf{R}$  and  $\mathbf{S}$  cannot be jointly estimated with the remaining parametric matrices, implementing a two-stage estimation instead.

Equation (13) implies,

$$\mathbf{H}_i^\psi \Psi_f = \mathbf{Z}_f - \mathbf{O}_i \mathbf{X}_f - \mathbf{H}_i^u \mathbf{U}_f \quad (28)$$

where the states,  $\mathbf{X}_f$ , will be replaced with the estimates given by (17). The noise term  $\mathbf{H}_i^\psi \Psi_f$  has a particular structure, with one-step-ahead errors in the first row, two-step-ahead errors in the second row and so on. Under these conditions, the gaussian loglikelihood function can then be written as,

$$\log l(\boldsymbol{\beta}) = -\frac{im}{2} \log(2\pi) - \frac{i}{2} \log(|\boldsymbol{\Sigma}|) - \frac{1}{2} \text{tr}(\Psi_f' \mathbf{H}_i^{\psi'} \boldsymbol{\Sigma}^{-1} \mathbf{H}_i^\psi \Psi_f) \quad (29)$$

where,  $\text{tr}(\cdot)$  is the trace operator and  $\boldsymbol{\Sigma}$  is the Prediction Error (PE) covariance. The noise term is obviously autocorrelated, so  $\boldsymbol{\Sigma}$  has the following structure,

$$\boldsymbol{\Sigma} = \mathbf{O}_i \mathbf{P}_i \mathbf{O}_i' + \mathbf{H}_i^\psi (\mathbf{I}_i \otimes \mathbf{Q}) \mathbf{H}_i^{\psi'} \quad (30)$$

where  $\otimes$  is the Kronecker product and  $\mathbf{P}_i$  is the state covariance matrix. There are two components in (30): the first addend in the right-hand-side refers to the error covariance of the states, while the second addend corresponds to the future output error variance, conditional to the estimated states. Taking into account the structure of  $\mathbf{H}_i^\psi \Psi_f$ , its covariance matrix

can be defined as:

$$\boldsymbol{\Sigma} = \begin{cases} \Sigma_{jk} \equiv PE_j \text{ covariance matrix} & \text{when } j = k \\ \Sigma_{jk} \equiv \text{cov}(PE_j, PE_k) & \text{when } j \neq k \end{cases} \quad j, k = 1, 2, \dots, i \quad (31)$$

where  $PE_j$  denotes the  $j$ -step-ahead prediction errors. Computing expression (30) requires, therefore, propagating  $i$  times the Kalman filter covariance equations (27) as in Subsection 3.2.2, to get an estimate of  $\mathbf{P}_i$ .

## 4 Montecarlo experiments

We will now analyze the performance of SUBEST1 and SUBEST2 in finite samples using as benchmark a SS based ML algorithm (Casals et al., 1999) initialized with the true parameter values. On the other hand, SUBEST1 and SUBEST2 will always be initialized with  $\boldsymbol{\beta} = \mathbf{0}$  and a value of  $i$  heuristically chosen as the nearest integer to  $\log T$ , where  $T$  is the sample size. Obviously, these settings deliberately favor the ML algorithm.

Tables 1-8 summarize the main results. The simulations in Tables 1-6 refer to different homoskedastic models. For each data generating process, we have obtained 1,000 samples with  $T = 50$  and  $T = 300$  observations, after discarding the first 50 values in each case to improve randomization. On the other hand, Tables 7-8 show the results obtained for two common conditional heteroskedastic models. In these cases the sample sizes are increased to  $T = 500$  and  $T = 3,000$ , to make them representative of the high

frequency series to which these models are typically applied.

Tables 1-2 show the results obtained assuming gaussian AR(2), ARMA(2,1) and bivariate VARMA(2,1) data generating processes, respectively. In all cases the AR parameters were chosen so that the roots are complex and far from the unit circle. This avoids ill-conditioning situations due to approximate cancellation of real AR and MA roots. As could be expected:

1. The averages of ML estimates are closer to the true parameter values for models with MA terms. However, if the model is pure AR, the root mean-squared errors (RMSEs) of SUBEST2 estimates are comparable to those of ML and generally better than those of SUBEST1.
2. Computational cost of ML ranges from 3.5-4 and 8-10 times the cost of SUBEST1 for  $T = 50$  and  $T = 300$ , respectively. On the other hand, the cost of SUBEST2 typically doubles that of SUBEST1 for  $T = 50$ . This overhead decreases when the sample size grows.

[INSERT TABLES 1-2]

Table 3 shows the results corresponding to a gaussian ARMA(1,1) $\times(0, 1)_s$  process with quarterly ( $s = 4$ ) and monthly data ( $s = 12$ ). We deliberately deteriorated the conditioning of the loss function by choosing parameter values that generate a partial redundancy between the roots of the regular AR and MA factors. Note that ML keeps an advantage in precision, but its computational overhead multiplies the cost of SUBEST1 by a 4-6 factor in the quarterly model and a 9-14 factor in the monthly model. This is due to

the high dimension of the state vector, which also affects the relative performance of SUBEST2, and to the ill-conditioning of the loss function. With  $T = 300$ , ML and SUBEST2 provide similar results in terms of accuracy.

[INSERT TABLE 3]

The literature shows that, in small samples, ML estimates display a “Pile-Up Effect” (PUE). This effect means that the distribution of ML estimates of moving average terms is multimodal, being some modes in the invertibility boundary, see Ansley and Newbold (1980). To further analyze the properties of the estimation methods, we therefore need more details about this PUE. Table 4 shows the results of this analysis, which main conclusions are: i) SUBEST1 and SUBEST2 present less PUE than ML, ii) discounting the PUE, estimates of the MA parameters are very similar for SUBEST2 and ML, and iii) SUBEST1 and SUBEST2 show slower convergence to the true MA seasonal parameters when the seasonal frequency is large, e.g.  $s = 12$ .

[INSERT TABLE 4]

The state-space foundation of the methods proposed makes them very flexible, in the sense that they can accommodate a wide class of models. To illustrate this flexibility, Tables 5-6 and 7-8 show, respectively, the results obtained with two unobserved variable models and two conditionally heteroscedastic formulations.

Specifically, Tables 5 and 6 show the results obtained with two typical formulations in the SS literature: an STSM with a low signal-to-noise ratio

and an AR(2) model with observation errors, respectively. Table 6 shows that in this case SUBEST1 is, not only very imprecise, but also the slower method. This is due to the fact that the loss function considered in SUBEST1 does not depend on the error covariance matrix. Therefore, its estimates do not take into account the null covariance restrictions. Also, the low signal-to-noise ratio makes the matter worst, as it deteriorates the conditioning of the likelihood surface, inducing convergence problems. On the other hand SUBEST2 and ML estimates are more precise, as they take into account this independence restriction, and their computational overhead is also similar, due to the very small number of parameters to be estimated. Results provided by the Table 6 are rather similar, being the main difference the improvement of SUBEST1 performance due to a higher signal-to-noise ratio.

[INSERT TABLES 5-6]

Finally, Tables 7 and 8 present the results for two common high frequency financial models: Autoregressive Stochastic Volatility (ARSV, see Harvey et al., 1994), and Generalized Autoregressive Conditional Heteroskedasticity (GARCH, see Bollerslev, 1986), models.

Table 7 displays an ARSV(1) model, where  $z_t$  are the observed returns and  $\sigma_t$  are the corresponding volatilities. In this type of models, the variance of  $\epsilon_t$  is assumed to be known and, therefore, it is not estimated. Results of the simulation are close to those of an AR with observation errors, because the SS representation of both models is very similar. However, when the sample size becomes larger, the computational cost of ML increases to 5-20 times

the cost of SUBEST1 and 3-12 times the cost of SUBEST2, for  $T = 500$  and  $T = 3,000$ , respectively.

Table 8 shows that the computational disadvantage of ML in the GARCH model is larger than in the ARSV. In particular, the load of ML is 49 times the cost of SUBEST1 and more than 16 times the cost of SUBEST2 for  $T = 500$ . When the sample size increases ( $T = 3,000$ ) SUBEST2 becomes the fastest method. In fact, ML converges 90 and 79 times slower than SUBEST2 and SUBEST1, respectively, while the three methods provide very similar results in terms of precision.

[INSERT TABLES 7-8]

## 5 Concluding Remarks

We presented two new algorithms to estimate time series models. These procedures are fast, stable and flexible, as they can be applied to any model written in its equivalent SS form.

Simulation results characterize the situations where each algorithm is more adequate. SUBEST1 is a good choice when the model does not include seasonal moving average terms and, as it is very fast, also when one wants to compute initial conditions estimates for a posterior ML iteration. On the other hand, SUBEST2 is more suitable as final estimation method, because its estimates are comparable to those of ML for large gaussian samples, while maintaining a substantial advantage in terms of both, computational

cost and numerical stability.

SUBEST1 and SUBEST2 may particularly useful when modeling high frequency financial data. The computational cost and instability of ML, as well as the fact that samples are far from the Gaussian assumption, strongly motivates the use of alternative estimation methods.

These algorithms are implemented in a MATLAB toolbox for time series modeling called  $E^4$ . The source code of this toolbox is freely provided under the terms of the GNU General Public License and can be downloaded at [www.ucm.es/info/icae/e4](http://www.ucm.es/info/icae/e4). This site also includes a complete user manual and other reference materials.

## References

- Ansley, C. F. and Newbold, P. (1980). Finite sample properties of estimators for autoregression moving average models. *Journal of Econometrics*, 13:159–183.
- Bauer, D. (2005). Estimating linear dynamical systems using subspace methods. *Econometric Theory*, 21:181–211.
- Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31:307–327.
- Casals, J., Sotoca, S., and Jerez, M. (1999). A fast stable method to compute the likelihood of time invariant state space models. *Economics Letters*, 65(3):329–337.
- Deistler, M., Peternell, K., and Scherrer, W. (1995). Consistency and relative efficiency of subspace methods. *Automatica*, 31(12):1865–1875.
- Dennis, J. E. and Schnabel, R. B. (1983). *Numerical methods for unconstrained optimization and nonlinear equations*. Prentice Hall, Englewood Cliffs, NJ.
- Dufour, J. M. and Pelletier, D. (2004). Linear estimation of weak VARMA models with macroeconomic applications. Technical report, Centre Interuniversitaire de Recherche en Economie Quantitative (CIREQ), Université de Montréal, CANADA.
- Favoreel, W., De Moor, B., and Van Overschee, P. (2000). Subspace state

- space system identification for industrial processes. *Journal of Process Control*, 10:149–155.
- Flores, R. and Serrano, G. (2002). A generalized least squares estimation method for VARMA models. *Statistics*, 36(4):303–316.
- Francq, C. and Zakoïan, J. M. (2000). Estimating weak GARCH representations. *Econometric Theory*, 16:692–728.
- Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computations*. Johns Hopkins University Press, Baltimore, third edition.
- Hannan, E. J. and Deistler, M. (1988). *The Statistical Theory of Linear Systems*. John Wiley, New York.
- Harvey, A. C. (1989). *Forecasting, structural time series models and the Kalman Filter*. Cambridge University Press.
- Harvey, A. C., Ruiz, E., and Shepard, N. (1994). Multivariate stochastic variance models. *Review of Economic Studies*, 61:247–264.
- Katayama, T. (2005). *Subspace Methods for System Identification*. Springer Verlag, London.
- Koreisha, S. G. and Pukkila, T. H. (1989). Fast linear estimation methods for vector autoregressive moving average models. *Journal of Time Series Analysis*, 10:325–339.
- Koreisha, S. G. and Pukkila, T. H. (1990). A generalized least-squares approach for estimation of autoregressive moving-average models. *Journal of Time Series Analysis*, 11:139–151.

- Ljung, L. (1999). *System Identification, Theory for the User*. PTR Prentice Hall, New Jersey, second edition.
- Lütkepohl, H. and Poskitt, D. S. (1996). Specification of echelon form VARMA models. *Journal of Business and Economic Statistics*, 14(1):69–79.
- Spliid, H. (1983). A fast estimation method for the vector autoregressive moving average model with exogenous variables. *Journal of the American Statistical Association*, 78(384):843–849.
- Van Overschee, P. and De Moor, B. (1994). N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93.

## Appendix A

From equation (13), displacing time subscripts, we obtain:

$$\mathbf{Z}_{f+} = \mathbf{O}_{i-1}\mathbf{X}_{i+1} + \mathbf{H}_{i-1}^u\mathbf{U}_{f+} + \mathbf{H}_{i-1}^\psi\mathbf{\Psi}_{f+} \quad (\text{A.1})$$

Note that  $\mathbf{Z}_{f+}$  includes only the future block information while  $\mathbf{Z}_f$  integrates the present and the future blocks. On the other hand, substituting (12b) into (12a) and propagating the state equation yields:

$$\mathbf{x}_t = (\Phi - \mathbf{E}\mathbf{H})^i\mathbf{x}_{t-i} + \sum_{j=0}^{i-1} (\Phi - \mathbf{E}\mathbf{H})^j [\mathbf{E}\mathbf{z}_{t-1-j} + (\Gamma - \mathbf{E}\mathbf{D})\mathbf{u}_{t-1-j}] \quad (\text{A.2})$$

which can be written in subspace form as,

$$\mathbf{X}_i = (\Phi - \mathbf{E}\mathbf{H})^i\mathbf{X}_p + \Xi_i(\Phi - \mathbf{E}\mathbf{H}, \mathbf{E})\mathbf{Z}_p + \Xi_i(\Phi - \mathbf{E}\mathbf{H}, \Gamma - \mathbf{E}\mathbf{D})\mathbf{U}_p \quad (\text{A.3})$$

where  $\Xi_i(\mathbf{A}, \mathbf{B}) = [\mathbf{A}^{i-1}\mathbf{B} \quad \mathbf{A}^{i-2}\mathbf{B} \quad \dots \quad \mathbf{A}\mathbf{B} \quad \mathbf{B}]$ , for any matrices  $\mathbf{A}, \mathbf{B}$ .

Again, displacing time indices, we obtain:

$$\begin{aligned} \mathbf{X}_{i+1} &= (\Phi - \mathbf{E}\mathbf{H})^{i+1}\mathbf{X}_p + \Xi_{i+1}(\Phi - \mathbf{E}\mathbf{H}, \mathbf{E})\mathbf{Z}_p \\ &+ \Xi_{i+1}(\Phi - \mathbf{E}\mathbf{H}, \Gamma - \mathbf{E}\mathbf{D})\mathbf{U}_p \end{aligned} \quad (\text{A.4})$$

or, in the same way,

$$\mathbf{X}_{i+1} = (\Phi - \mathbf{E}\mathbf{H})\mathbf{X}_i + \mathbf{E}\mathbf{Z}_{pr} + (\Gamma - \mathbf{E}\mathbf{D})\mathbf{U}_{pr} \quad (\text{A.5})$$

Finally, substituting (A.5) into (A.1), we get (18).

## Appendix B

To obtain the initial conditions  $\bar{\mathbf{K}}$  and  $\bar{\mathbf{B}}$ , we use the state equation of the Kalman filter, that can be written in subspace form as:

$$\hat{\mathbf{X}}_{i+1} = \hat{\Phi} \hat{\mathbf{X}}_i + \hat{\Gamma} U_{pr} + \mathbf{K}_i \tilde{\mathbf{Z}}_{pr} \quad (\text{B.1})$$

where  $\hat{\Phi}$ ,  $\hat{\Gamma}$  and  $\hat{\mathbf{X}}_i$  are previously estimated in (25) and the residual vector,  $\tilde{\mathbf{Z}}_{pr}$ , is exactly as in (24). Also, as in equation (17), the state sequence  $\hat{\mathbf{X}}_{i+1}$  is computed as:

$$\hat{\mathbf{X}}_{i+1} = \mathbf{O}_{i-1}^\dagger (\mathbf{Z}_{f+} \Pi_{U, \mathbf{Z}_{p+}} - \mathbf{H}_{i-1}^u \mathbf{U}_{f+}) \quad (\text{B.2})$$

Thus, the only unknown matrix is  $\mathbf{K}_i$ . From (B.1) we get  $\bar{\mathbf{K}}$ , an approximation for  $\mathbf{K}_i$ , as:

$$\bar{\mathbf{K}} = (\hat{\mathbf{X}}_{i+1} - \hat{\Phi} \hat{\mathbf{X}}_i - \hat{\Gamma} U_{pr}) \tilde{\mathbf{Z}}_{pr}^\dagger \quad (\text{B.3})$$

Note that  $\bar{\mathbf{K}}$  is a least square estimation of the Kalman filter gain from equation (B.1).

Finally, an approximation of  $\mathbf{B}_i$  is obtained as,  $\bar{\mathbf{B}} = (T - 2i + 1)^{-1} \tilde{\mathbf{Z}}_{pr} \tilde{\mathbf{Z}}_{pr}'$ .

## Tables

Table 1<sup>†</sup>: Univariate nonseasonal models with gaussian errors.

Table 1.1 AR(2) model:  $(1-.4B+.3B^2)z_t = a_t; a_t \sim iidN(0, 1.0)$ .

Method		SUBEST1			SUBEST2			ML		
$T$	True values	-.4	.3	1.0	-.4	.3	1.0	-.4	.3	1.0
50	Average	-.384	.305	.931	-.367	.284	.943	-.386	.304	.946
	Std. Dev	.144	.144	.203	.136	.127	.198	.137	.135	.196
	RMSE	.145	.144	.203	<u>.136</u>	<u>.127</u>	.198	.138	.135	<u>.196</u>
	Time	100%			216%			416%		
300	Average	-.397	.306	.985	-.391	.296	.992	-.395	.300	.994
	Std. Dev	.059	.066	.083	.053	.055	.082	.053	.056	.082
	RMSE	.059	.066	<u>.083</u>	<u>.054</u>	<u>.055</u>	<u>.083</u>	<u>.054</u>	.065	<u>.083</u>
	Time	100%			219%			1040%		

Table 1.2 ARMA(2,1) model:  $(1-.4B+.3B^2)z_t = (1-.8B)a_t; a_t \sim iidN(0, 1.0)$ .

Method		SUBEST1				SUBEST2				ML			
$T$	True values	-.4	.3	-.8	1.0	-.4	.3	-.8	1.0	-.4	.3	-.8	1.0
50	Average	-.360	.289	-.671	1.020	-.307	.297	-.680	1.000	-.354	.313	-.800	.927
	Std. Dev	.305	.175	.291	.214	.222	.141	.254	.206	.196	.141	.197	.194
	RMSE	.308	.175	.318	.215	.240	<u>.141</u>	.281	<u>.206</u>	<u>.201</u>	<u>.141</u>	<u>.197</u>	<u>.207</u>
	Time	100%				244%				359%			
300	Average	-.398	.309	-.763	1.014	-.389	.304	-.805	1.001	-.395	.304	-.797	.993
	Std. Dev	.085	.071	.061	.085	.066	.061	.074	.083	.066	.061	.050	.081
	RMSE	.085	.072	.072	.086	.067	<u>.062</u>	.074	.083	<u>.066</u>	<u>.062</u>	<u>.050</u>	<u>.081</u>
	Time	100%				230%				881%			

<sup>†</sup>ML algorithm is initialized with the true parameter values. RMSE is the root mean-squared error. The smallest RMSEs are underlined.

Table 2<sup>†</sup>: Bivariate VARMA model with gaussian errors.

$$\begin{pmatrix} 1 - .70B + .60B^2 & 0 \\ 0 & 1 - 1.30B + .50B^2 \end{pmatrix} \begin{pmatrix} z_{1t} \\ z_{2t} \end{pmatrix} = \\ \begin{pmatrix} 1 - .30B & -.90B \\ .60B & 1 - .80B \end{pmatrix} \begin{pmatrix} a_{1t} \\ a_{2t} \end{pmatrix} \\ \begin{pmatrix} a_{1t} \\ a_{2t} \end{pmatrix} \sim iidN \left[ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} .07 & .02 \\ - & .05 \end{pmatrix} \right]$$

Table 2.1: Estimation results with sample size  $T = 50$

True values	- .70	-1.30	.60	.50	-.30	.60	-.90	-.80	.07	.02	.05
Method	SUBEST1										
Average	-.671	-1.359	.545	.557	-.267	.431	-.743	-.736	.079	.019	.056
Std. Dev	.137	.233	.104	.177	.200	.104	.177	.246	.017	.010	.014
RMSE	.140	.240	.118	.186	.203	.199	.236	.254	.019	<u>.010</u>	.015
Time	100%										
Method	SUBEST2										
Average	-.620	-1.297	.519	.500	-.331	.453	-.832	-.776	.073	.017	.056
Std. Dev	.133	.202	.100	.151	.208	.121	.195	.251	.016	.010	.014
RMSE	.155	.202	.129	<u>.151</u>	.210	.190	<u>.207</u>	.252	<u>.017</u>	<u>.010</u>	.015
Time	348%										
Method	ML										
Average	-.686	-1.279	.589	.492	-.344	.654	-1.024	-.872	.060	.017	.044
Std. Dev	.087	.188	.072	.155	.175	.122	.179	.220	.014	.010	.010
RMSE	<u>.088</u>	<u>.189</u>	<u>.073</u>	.155	<u>.181</u>	<u>.134</u>	.218	<u>.231</u>	<u>.017</u>	<u>.010</u>	<u>.012</u>
Time	398%										

<sup>†</sup>ML algorithm is initialized with the true parameter values. RMSE is the root mean-squared error. The smallest RMSEs are underlined.

Table 2.2: Estimation results with sample size  $T = 300$

True values	-0.70	-1.30	0.60	0.50	-0.30	0.60	-0.90	-0.80	0.07	0.02	0.05
Method	SUBEST1										
Average	-0.693	-1.318	0.580	0.515	-0.276	0.536	-0.829	-0.769	0.074	0.021	0.054
Std. Dev	0.044	0.090	0.038	0.072	0.084	0.044	0.063	0.099	0.006	0.004	0.005
RMSE	0.045	0.092	0.043	0.074	0.087	0.078	0.094	0.104	0.008	<u>0.004</u>	0.006
Time	100%										
Method	SUBEST2										
Average	-0.686	-1.311	0.584	0.506	-0.289	0.585	-0.902	-0.831	0.072	0.019	0.053
Std. Dev	0.039	0.076	0.034	0.062	0.070	0.043	0.072	0.089	0.006	0.004	0.005
RMSE	0.042	0.076	0.037	0.062	0.070	0.046	0.072	0.095	<u>0.006</u>	<u>0.004</u>	0.006
Time	342%										
Method	ML										
Average	-0.699	-1.296	0.599	0.500	-0.305	0.605	-0.913	-0.804	0.069	0.019	0.049
Std. Dev	0.030	0.055	0.023	0.047	0.062	0.035	0.052	0.066	0.006	0.004	0.004
RMSE	<u>0.030</u>	<u>0.055</u>	<u>0.023</u>	<u>0.047</u>	<u>0.062</u>	<u>0.035</u>	<u>0.054</u>	<u>0.066</u>	<u>0.006</u>	<u>0.004</u>	<u>0.004</u>
Time	744%										

Table 3<sup>†</sup>: Univariate seasonal ARMA models.

Table 3.1 ARMA(1, 1)  $\times$  (0, 1)<sub>4</sub> model:  $(1 - .4B)z_t = (1 - .7B)(1 - .8B^4)a_t$ ;  
 $a_t \sim iidN(0, 1.0)$

Method		SUBEST1				SUBEST2				ML			
$T$	True Values	-.4	-.7	-.8	1.0	-.4	-.7	-.8	1.0	-.4	-.7	-.8	1.0
50	Average	-.208	-.390	-.456	1.174	-.294	-.632	-.742	.924	-.325	-.683	-.835	.886
	Std. Dev	.557	.585	.168	.336	.486	.482	.153	.210	.325	.329	.169	.204
	RMSE	.589	.662	.383	.378	.497	.486	<u>.163</u>	<u>.223</u>	<u>.333</u>	<u>.330</u>	.173	<u>.233</u>
	Time	100%				217%				441%			
300	Average	-.302	-.591	-.575	1.228	-.365	-.665	-.826	.998	-.388	-.695	-.800	.990
	Std. Dev	.250	.267	.088	.153	.147	.132	.064	.084	.146	.122	.042	.080
	RMSE	.269	.288	.242	.275	.151	.136	.069	.084	<u>.146</u>	<u>.122</u>	<u>.042</u>	<u>.081</u>
	Time	100%				227%				577%			

Table 3.2 ARMA(1, 1)  $\times$  (0, 1)<sub>12</sub> model:  
 $(1 - .4B)z_t = (1 - .7B)(1 - .8B^{12})a_t$ ;  
 $a_t \sim iidN(0, 1.0)$

Method		SUBEST1				SUBEST2				ML			
$T$	True values	-.4	-.7	-.8	1.0	-.4	-.7	-.8	1.0	-.4	-.7	-.8	1.0
50	Average	-.240	-.493	-.368	1.133	-.244	-.592	-.494	1.073	-.324	-.680	-.807	.902
	Std. Dev	.525	.534	.128	.261	.438	.454	.042	.237	.331	.323	.224	.225
	RMSE	.549	.572	.450	.293	.465	.467	.309	.248	<u>.340</u>	<u>.323</u>	<u>.224</u>	<u>.245</u>
	Time	100%				436%				907%			
300	Average	-.374	-.661	-.586	1.221	-.379	-.691	-.812	.970	-.386	-.694	-.805	.984
	Std. Dev	.221	.201	.055	.118	.143	.128	.073	.084	.135	.112	.048	.083
	RMSE	.222	.205	.221	.250	.145	.128	.074	.089	<u>.136</u>	<u>.112</u>	<u>.048</u>	<u>.085</u>
	Time	100%				371%				1415%			

<sup>†</sup>ML algorithm is initialized with the true parameter values. RMSE is the root mean-squared error. The smallest RMSEs are underlined.

Table 4: Measuring the Pile-Up Effect (PUE) in small samples ( $T = 50$ ).

Table	Model	Parameter	% of PUE		Estimates discounting PUE		
			SUBEST1	SUBEST2	SUBEST1	SUBEST2	
1.1	ARMA(2,1)	$\theta_1 = -.8$	10.2	2.1	21.5	-671	-740
2.1	VARMA(2,1)	$\theta_{11} = -.3$	.0	.0	.0	-.267	-.344
2.1	VARMA(2,1)	$\theta_{22} = -.8$	10.2	15.9	26.6	-.695	-.777
3.1	ARMA(1,1) $\times$ (0,1) <sub>4</sub>	$\theta_1 = -.7$	.0	1.5	22.6	-.390	-.627
3.1	ARMA(1,1) $\times$ (0,1) <sub>4</sub>	$\Theta_1 = -.8$	.0	2.7	32.2	-.456	-.735
3.2	ARMA(1,1) $\times$ (0,1) <sub>12</sub>	$\theta_1 = -.7$	3.5	.6	23.9	-.472	-.590
3.2	ARMA(1,1) $\times$ (0,1) <sub>12</sub>	$\Theta_1 = -.8$	.0	.0	46.4	-.368	-.629

We consider PUE when a MA parameter estimate is lesser or equal to -1.00.

Table 5<sup>†</sup>: Structural time series model.

$$\begin{pmatrix} T_{t+1} \\ \Delta_{t+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} T_t \\ \Delta_t \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \eta_t$$

$$z_t = (1 \quad 0) \begin{pmatrix} T_t \\ \Delta_t \end{pmatrix} + \epsilon_t; \quad \begin{pmatrix} \eta_t \\ \epsilon_t \end{pmatrix} \sim iidN \left[ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} .01 & 0 \\ 0 & 10 \end{pmatrix} \right]$$

Method		SUBEST1		SUBEST2		ML	
$T$	True values	.01	10	.01	10	.01	10
50	Average	1.475	8.473	.022	9.726	.035	9.911
	Std. Dev	3.275	3.693	.040	1.406	.069	1.005
	RMSE	3.588	3.996	<u>.042</u>	1.432	.073	<u>1.009</u>
	Time	395%		125%		100%	
300	Average	.461	10.137	.019	9.977	.008	10.005
	Std. Dev	.146	.476	.014	.443	.006	.409
	RMSE	.474	.495	.017	.444	<u>.007</u>	<u>.409</u>
	Time	273%		100%		171%	

Table 6<sup>†</sup>: AR(2) model with observation errors.

$$(1 - 1.5B + .8B^2)z_t^* = a_t; \quad z_t = z_t^* + v_t;$$

$$\begin{pmatrix} a_t \\ v_t \end{pmatrix} \sim iidN \left[ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1.0 & 0 \\ 0 & .5 \end{pmatrix} \right]$$

Method		SUBEST1				SUBEST2				ML			
$T$	True values	-1.5	.8	1.0	.5	-1.5	.8	1.0	.5	-1.5	.8	1.0	.5
50	Average	-1.425	.731	1.030	.558	-1.429	.737	1.045	.558	-1.480	.781	.954	.469
	Std. Dev	.268	.187	.219	.202	.133	.121	.195	.177	.128	.119	.185	.169
	RMSE	.278	.199	.221	.210	.150	.136	.200	.186	<u>.129</u>	<u>.121</u>	<u>.190</u>	<u>.172</u>
	Time	100%				204%				142%			
300	Average	-1.491	.798	.990	.530	-1.487	.788	1.010	.513	-1.492	.793	.997	.493
	Std. Dev	.057	.046	.081	.069	.047	.044	.075	.057	.044	.043	.072	.055
	RMSE	.057	.047	.081	.076	.048	.046	.076	.059	<u>.045</u>	<u>.043</u>	<u>.072</u>	<u>.056</u>
	Time	100%				193%				253%			

<sup>†</sup>ML algorithm is initialized with the true parameter values. RMSE is the root mean-squared error. The smallest RMSEs are underlined.

Table 7<sup>†</sup>: ARSV(1) model in State Space form.

$$\begin{aligned} z_t &= 1.5\sigma_t\epsilon_t \\ \log(\sigma_{t+1}^2) &= .95\log(\sigma_t^2) + \eta_t \end{aligned}$$

where  $\eta_t \sim iidN(0, .5)$ ,  $\epsilon_t \sim iidN(0, 1)$  and,  $\epsilon_t$  and  $\eta_t$  are independent.

Method		SUBEST1			SUBEST2			ML		
$T$	True values	.95	1.50	.50	.95	1.50	.50	.95	1.50	.50
500	Average	.932	1.414	.469	.945	1.453	.462	.933	1.471	.517
	Std. Dev	.164	1.105	.146	.041	1.027	.146	.034	.460	.104
	RMSE	.165	1.108	.149	.041	1.028	.151	<u>.038</u>	<u>.461</u>	<u>.106</u>
	Time	100%			168%			476%		
3000	Average	.949	1.485	.497	.949	1.484	.494	.948	1.490	.501
	Std. Dev	.010	.202	.044	.009	.198	.044	.009	.183	.038
	RMSE	.010	.202	.044	<u>.009</u>	.198	.044	<u>.009</u>	<u>.184</u>	<u>.039</u>
	Time	100%			166%			2059%		

Table 8<sup>†</sup>: GARCH(1,1) model in ARMA form.

$$\begin{aligned} y_t &= \epsilon_t \text{ with } \epsilon_t | \Omega_{t-1} \sim iidN(0, \sigma_t^2) \text{ where} \\ \epsilon_t^2 &= 1.0 + \frac{1-.80B}{1-.97B} v_t \text{ with } v_t = \epsilon_t^2 - \sigma_t^2 \end{aligned}$$

Method		SUBEST1			SUBEST2			ML		
$T$	True values	1.0	-.97	-.80	1.0	-.97	-.80	1.0	-.97	-.80
500	Average	1.005	-.972	-.798	1.002	-.969	-.814	1.209	-.952	-.784
	Std. Dev	.766	.025	.076	.760	.031	.106	1.420	.037	.056
	RMSE	.766	<u>.025</u>	.076	<u>.760</u>	.031	.107	1.436	.041	<u>.058</u>
	Time	100%			298%			4916%		
3000	Average	.999	-.974	-.810	.998	-.972	-.809	1.095	-.967	-.797
	Std. Dev	.314	.015	.052	.313	.019	.068	.880	.011	.018
	RMSE	.314	.016	.053	<u>.313</u>	.019	.069	.885	<u>.011</u>	<u>.018</u>
	Time	114%			100%			9047%		

<sup>†</sup>ML algorithm is initialized with the true parameter values. RMSE is the root mean-squared error. The smallest RMSEs are underlined.