

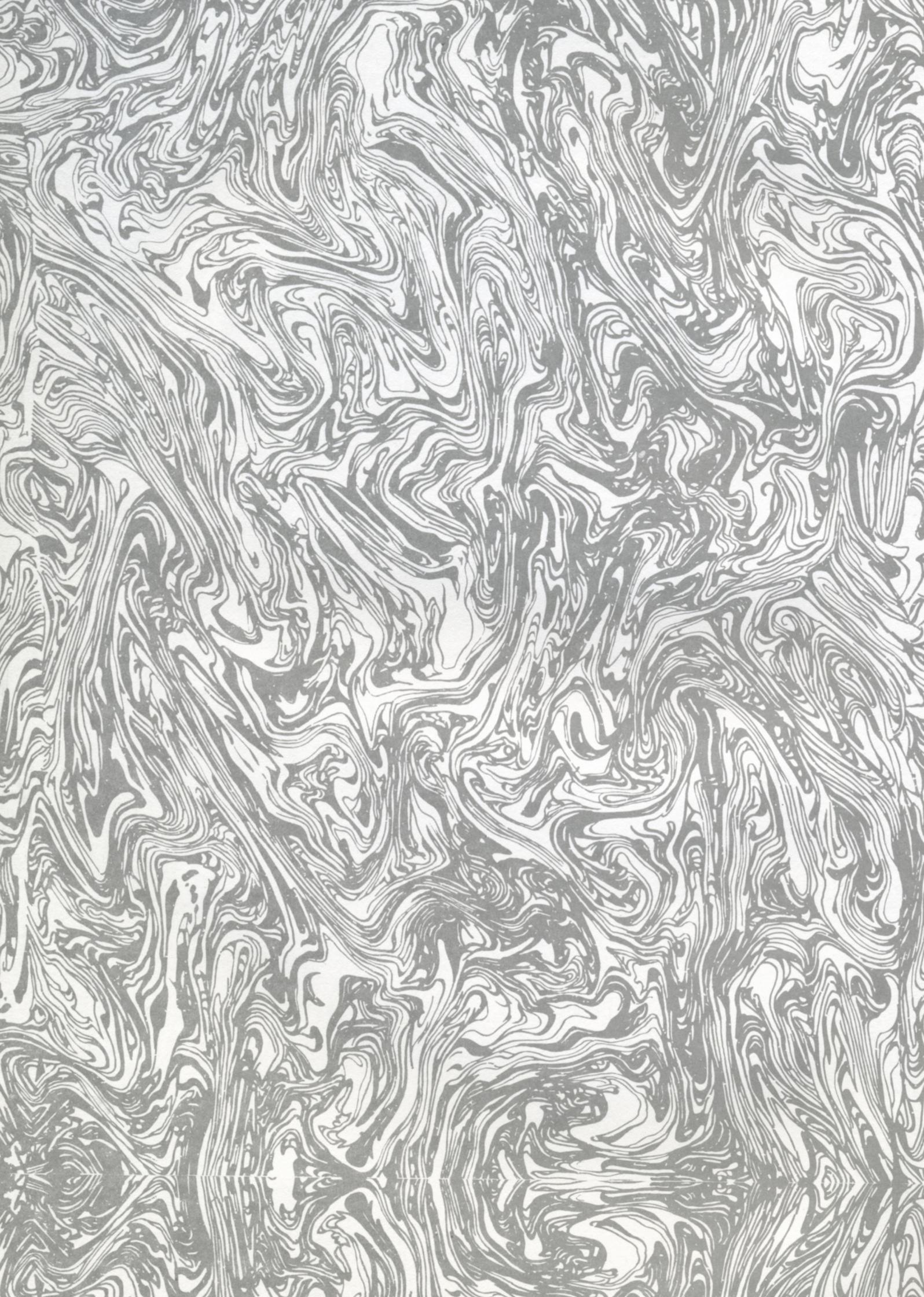
**EVALUACION
Y
MAXIMIZACION
DE LA
FUNCION DE VEROSIMILITUD
DE
PROCESOS ARMA MULTIVARIANTES**

TESIS DOCTORAL

**José Alberto Mauricio Arias
Director: Arthur B. Treadway**

**Departamento de Economía Cuantitativa
Facultad de Ciencias Económicas y Empresariales
Universidad Complutense de Madrid**

1992





Versión 1.1

Copyright © 1992, 2024 José Alberto Mauricio

Versión 1.0 (1992): Impresión original en papel

Versión 1.1 (2024): Reproducción en formato electrónico de la Versión 1.0

Versión 1.0 escaneada (2002) disponible en <https://hdl.handle.net/20.500.14352/63344>

ISBN 978-84-669-0137-6

Universidad Complutense de Madrid - Servicio de Publicaciones

**EVALUACION
Y
MAXIMIZACION
DE LA
FUNCION DE VEROSIMILITUD
DE
PROCESOS ARMA MULTIVARIANTES**

TESIS DOCTORAL

José Alberto Mauricio Arias

Director: **Arthur B. Treadway**

Departamento de Economía Cuantitativa
Facultad de Ciencias Económicas y Empresariales
Universidad Complutense de Madrid

1992

Dedicado a mi hermano Raúl

Agradecimientos

Deseo expresar mi más sincero agradecimiento al Profesor Arthur B. Treadway por su labor de dirección y, sobre todo, por darme la oportunidad de llevar a cabo una investigación como ésta. También quiero agradecer las valiosas sugerencias recibidas de los Profesores Miguel Jerez y Alfonso Novales, que, sin duda, han contribuido a mejorar la calidad global del trabajo.

Por otro lado, el apoyo de mis compañeros José Luis Gallego, Víctor M. Gonzalo, Alfredo Medinilla y Raquel del Río, junto con los buenos ratos pasados en su compañía, ha sido fundamental durante todo el proceso de elaboración de la tesis. También quiero agradecer su interés al resto de compañeros del Departamento que, de un modo u otro, han mostrado alguna inquietud por este trabajo.

Desde Enero de 1991, mi labor como ayudante en el *Servicio de Previsión y Seguimiento de la Economía Española*, patrocinado por la *Caja de Ahorros y Monte de Piedad de Madrid*, ha sido decisiva para dar un carácter marcadamente práctico a esta investigación y para obtener, durante los dos últimos años, un apoyo financiero nada despreciable. En este sentido, también quiero subrayar que mi *arma* de trabajo más preciada (un ordenador personal *HP Vectra 386/25*) ha sido financiada por el Proyecto de Investigación N° PB89-0129 de la DGICYT ("*Experimentos en la Econometría de Series Temporales*", dirigido por el Profesor Arthur B. Treadway). El apoyo prestado por estas dos instituciones merece mi más cordial agradecimiento.

Fuera del entorno profesional, la colaboración, en todos los sentidos, de mi familia (especialmente de mis padres) y de Rosa, ha sido, sin duda alguna, el apoyo más importante que he recibido durante el último año. Espero que todos los inconvenientes y la falta de atención que han tenido que sufrir, a causa de mi dedicación a este trabajo, hayan merecido la pena.

Madrid

Noviembre de 1992

J.A.M.

Indice

Capítulo 1: Introducción	1
Capítulo 2: Evaluación de la función de verosimilitud exacta de procesos ARMA estacionarios	5
2.1. Formulación de los procesos ARMA estacionarios	6
2.1.1. Procesos ARMA(ρ, q) multivariantes estocásticos	7
2.1.2. Formulación compacta de procesos ARMA(ρ, q) multivariantes estocásticos	9
2.1.3. Referencias y notas para la sección 2.1	12
2.2. Repaso de procedimientos existentes	13
2.2.1. Evaluación directa de la función de verosimilitud exacta	15
2.2.2. La formulación de Nicholls y Hall (1979)	20
2.2.3. Técnicas basadas en las representaciones en <i>espacio de los estados</i>	25
2.2.4. La aproximación de Hillmer y Tiao (1979)	28
2.3. Un nuevo procedimiento para evaluar la función de verosimilitud exacta	29
2.3.1. El algoritmo [FVE]	29
2.3.2. Algunos aspectos computacionales	38
2.3.3. Notas para la sección 2.3	42
2.4. Análisis comparativo de distintos procedimientos	43
Capítulo 3: Estimación de procesos ARMA estacionarios por máxima verosimilitud	49
3.1. Estimación por máxima verosimilitud exacta	50
3.1.1. Formulación de la función objetivo	51
3.1.2. Descripción del procedimiento de optimización	53
3.1.3. Análisis y aplicaciones del procedimiento de estimación	58
3.1.4. Notas para la sección 3.1	70
3.2. Otros procedimientos de estimación	73
3.2.1. Estimación por máxima verosimilitud aproximada	75
3.2.2. Estimación por máxima verosimilitud condicional	79
3.3. Análisis comparativo de distintos procedimientos de estimación	87
Capítulo 4: Conclusiones	105
Apéndice A.1: Resultados de álgebra matricial	111

Apéndice A.2: Codificación del algoritmo [FVE]	114
Introducción	114
Algoritmo LUDCMP	116
Algoritmo LUSOL	118
Algoritmo CHOLDCMP	119
Algoritmo CHOLFOR	121
Algoritmo CHOLBACK	122
Algoritmo CGAMMA	122
Algoritmo CXI	125
Algoritmo CRES	127
Algoritmo FVE	129
Apéndice A.3: Codificación del algoritmo [MAXFV]	137
Introducción	137
Algoritmo CEPSMQ	138
Algoritmo DEFPAR	138
Algoritmo CFOBJ	141
Algoritmo CGRADDC	142
Algoritmo CLP	143
Algoritmo ACTBFGS	145
Algoritmo CNBFGS	147
Algoritmo CONVO	148
Algoritmo CONV	149
Algoritmo MAXFVCN	150
Algoritmo MAXFV	153
Apéndice A.4: El vector gradiente de la función objetivo	156
Apéndice A.5: Simulación de procesos ARMA multivariantes	162
Referencias bibliográficas	165

CAPITULO 1

Introducción

El objetivo de este trabajo consiste en la descripción teórica, la puesta en práctica y el estudio empírico del funcionamiento de un nuevo algoritmo para la estimación por máxima verosimilitud de modelos ARMA multivariantes estocásticos. El problema de la estimación se estudia diferenciando claramente entre *evaluación* y *maximización* de la función de verosimilitud. Esta distinción permite analizar en detalle, integrar adecuadamente y sacar el máximo partido de todos los elementos que componen el problema global de la estimación. Además, los problemas asociados a cada una de esas dos tareas son tan diferentes entre sí, que su estudio por separado resulta imprescindible si se desea prestar la debida atención a los aspectos críticos de su utilización en la práctica.

Esta distinción entre evaluación y maximización de la función de verosimilitud no ha recibido mucha atención en la literatura sobre estimación de modelos ARMA. Si bien existe una gran cantidad de trabajos teóricos sobre la evaluación de la función de verosimilitud [por ejemplo Hillmer y Tiao (1979), Nicholls y Hall (1979), Hall y Nicholls (1980) y Shea (1984, 1987)], son muy escasas las referencias que prestan atención al problema de su maximización. Además, dicha atención suele limitarse a sugerir en unas pocas líneas el empleo de algún método de optimización estándar para maximizar la función de verosimilitud evaluada según se describe ampliamente en dichos trabajos [*vid.* por ejemplo Hillmer y Tiao (1979) y Hall y Nicholls (1980)]. Una interesante excepción es el trabajo de Shea (1984).

Por otro lado, aunque el problema de la evaluación de la función de verosimilitud ha recibido muchísima atención desde la publicación del libro de Box y Jenkins (1970), las diferentes alternativas existentes en la actualidad no pueden considerarse, desde nuestro punto de vista, plenamente satisfactorias. Esto es debido a que todas las propiedades que, en nuestra opinión, debe reunir un algoritmo para llevar a cabo dicha evaluación se encuentran dispersas entre los muchos trabajos existentes sobre el tema, de manera que si bien ciertos procedimientos poseen una serie de características interesantes, carecen de otras que sí pueden encontrarse en procedimientos alternativos.

Así, aunque los algoritmos de Shea (1984, 1987) [incorporados en la subrutina G13DCF de la librería *N.A.G.* (1987)] pueden considerarse los más eficientes desde el punto de vista computacional

entre los disponibles actualmente, su empleo no permite detectar automáticamente situaciones de no invertibilidad ni evaluar el vector de residuos *analíticamente exactos* correspondientes a la muestra y a los valores numéricos de los parámetros considerados. Estas tareas sí tienen cabida en el algoritmo de Hall y Nicholls (1980) y en la extensión al contexto multivariante del procedimiento de Ljung y Box (1979). No obstante, el primero implica un elevado coste computacional y el segundo cierta ineficiencia y pérdida de precisión numérica, debidas a la necesidad de evaluar explícitamente la inversa de una matriz. Por último, el procedimiento de Hillmer y Tiao (1979) [ofrecido por *The S.C.A. Statistical System*; vid. Liu y Hudak (1986, 1992)] no permite, para procesos con parte autorregresiva, evaluar ni la función de verosimilitud ni el vector de residuos de forma analíticamente exacta, lo cual puede suponer un grave inconveniente cuando, por ejemplo, la muestra considerada contiene anomalías importantes dentro de las primeras observaciones.

Teniendo en cuenta este panorama, en el presente trabajo se diseña un procedimiento computacionalmente eficiente para estimar por máxima verosimilitud procesos ARMA multivariantes, integrando, tras un análisis detallado por separado, los problemas de la evaluación y posterior maximización de la función de verosimilitud. El nuevo procedimiento que se propone como alternativa a los disponibles actualmente está basado en argumentos teóricos rigurosos y proporciona en la práctica:

- [1] Buenas estimaciones de los parámetros en situaciones tanto bien como mal condicionadas (muestras cortas, parámetros cercanos a la no estacionariedad o a la no invertibilidad, presencia de observaciones influyentes, ...).
- [2] Indicadores automáticos de especificaciones inadecuadas (no estacionarias o no invertibles) así como instrumentos (residuos analíticamente exactos) para efectuar la diagnosis de los modelos estimados.

El algoritmo que se propone en este trabajo parte de un profundo análisis de las ideas contenidas en los procedimientos disponibles actualmente. De esta forma, se pueden descubrir y explotar nuevas posibilidades ignoradas en dichos trabajos. Este análisis, junto con ciertas innovaciones, permite recoger en un sólo algoritmo las ventajas que se encuentran por separado en los procedimientos disponibles actualmente, a la vez que permite demostrar que dichos algoritmos son simplemente un caso particular (en ocasiones, arbitrariamente particular) del procedimiento descrito en este trabajo. Con el fin de aprovechar efectivamente las ventajas que dicho algoritmo puede ofrecer, a lo largo del trabajo se describen tanto su estructura teórica básica como los detalles de su puesta en práctica (algo habitualmente ignorado en los trabajos teóricos sobre el tema).

La exposición que se utiliza en el trabajo, encaminada a conseguir los objetivos mencionados, está organizada sobre dos pilares fundamentales:

- [1] La descripción *teórica* (con instrumentos matemáticos y estadísticos sencillos) y *codificada* de un nuevo procedimiento para evaluar y maximizar eficientemente la función de verosimilitud de procesos ARMA multivariantes, así como la comparación (teórica y empírica) de dicho procedimiento con los algoritmos disponibles en la actualidad.
- [2] La comprobación del correcto funcionamiento en la práctica de los métodos propuestos, mediante un amplio conjunto de ejercicios empíricos con datos tanto simulados como reales.

El trabajo está estructurado en cuatro capítulos (el primero de los cuales es esta introducción) y cinco apéndices, cuyo contenido se repasa brevemente a continuación.

Los Capítulos 2 y 3 constituyen la parte central del trabajo, tanto en extensión como en contenido. En concreto, en el Capítulo 2 se trata el tema de la evaluación de la función de verosimilitud *exacta* de procesos ARMA multivariantes estacionarios. Tras un resumen de la formulación y propiedades de dichos modelos, se revisan detenidamente los procedimientos más relevantes disponibles actualmente para tal fin. Partiendo de algunas sugerencias extraídas de este repaso, se describe un nuevo procedimiento para evaluar la función de verosimilitud exacta, que reúne todas las ventajas que se encuentran esparcidas entre los algoritmos revisados y carece de sus inconvenientes. El Capítulo 2 se cierra con un resumen comparativo que revela la potencial superioridad del algoritmo propuesto en este trabajo frente a los disponibles actualmente.

El Capítulo 3 se abre con la descripción detallada de un algoritmo, basado en algunas de las ideas sugeridas por Shea (1984), para maximizar la función de verosimilitud exacta evaluada según el algoritmo del Capítulo 2. También se contrasta el funcionamiento en la práctica del procedimiento de estimación resultante de la combinación de ambos algoritmos, mediante un conjunto de estimaciones en situaciones tanto simuladas como reales. A continuación, se estudian otros procedimientos de estimación derivados del de máxima verosimilitud exacta (incluidos los procedimientos basados en el criterio de *mínima suma de cuadrados*) y se comparan sus resultados con los proporcionados por dicho método. A modo de conclusión práctica, el Capítulo 3 se cierra con algunos ejemplos de situaciones en las que los métodos propuestos en este trabajo resultan más adecuados que los ofrecidos por uno de los programas comerciales más utilizados en el análisis de series temporales (*The S.C.A. Statistical System*).

Por último, en el Capítulo 4 se resumen las principales aportaciones del trabajo y se sugieren tanto posibles ampliaciones del mismo como futuras líneas de investigación apoyadas en él.

Por su parte, los Apéndices están estructurados de la siguiente manera. En el Apéndice A.1 se resumen algunos resultados de álgebra matricial, utilizados en el Capítulo 2 y en el Apéndice A.4, en el que se presenta un procedimiento para evaluar las primeras derivadas de la función de verosimilitud. Por último, en el Apéndice A.5 se resume la metodología empleada en los ejercicios de simulación realizados en el Capítulo 3.

Los Apéndices A.2 y A.3 merecen un comentario especial. Uno de los objetivos principales de este trabajo ha consistido en desarrollar procedimientos adecuados para ser empleados en la práctica econométrica habitual. Pensamos que la descripción teórica y el análisis empírico de dichos procedimientos no constituyen por sí solos, aunque se expongan muy detalladamente, materiales suficientes para lograr tal objetivo (aunque, por supuesto, sí necesarios). Por este motivo, se ha considerado fundamental incluir la descripción de los algoritmos propuestos de la forma más adecuada para su codificación en algún lenguaje de programación (como *Pascal* o *C*), de manera que su puesta en marcha resulte prácticamente inmediata. Con este propósito, en el Apéndice A.2 se describe en forma de *pseudocódigo* [vid. Dennis y Schnabel (1983)] el algoritmo propuesto en el Capítulo 2 para evaluar la función de verosimilitud exacta, mientras que en el Apéndice A.3 se describe el algoritmo propuesto en el Capítulo 3 para su maximización. La traducción del *pseudocódigo* a un lenguaje como *Pascal* o *C* resulta (para alguien familiarizado con la programación de ordenadores) muy sencilla, por lo que se dispone de un procedimiento de estimación completo que puede ponerse en marcha y utilizarse en la práctica inmediatamente.

Para concluir, puede señalarse que, si bien el objetivo de este trabajo no es la modelización de series temporales, está bastante claro que la estimación eficiente de un modelo es una parte crucial en dicho proceso. Por ello, pensamos que es importante disponer de métodos que proporcionen tanto las mejores estimaciones posibles como los instrumentos de diagnóstico más adecuados en cualquier situación, ya que las decisiones que se tomen sobre la posible reformulación de un modelo dependen críticamente de ambos factores. Los algoritmos que se proponen en este trabajo responden a dicha finalidad, por lo que pensamos que su empleo en la práctica econométrica puede resultar verdaderamente útil.

CAPITULO 2

Evaluación de la función de verosimilitud exacta de procesos ARMA estacionarios

El estudio de la función de verosimilitud exacta (*FVE* en adelante) de procesos ARMA estacionarios multivariantes, constituye tanto el punto de partida como el núcleo del presente trabajo. Como es bien sabido, la expresión matemática de la *FVE* puede escribirse de distintas formas. Dependiendo de los objetivos perseguidos, algunas de estas formas resultan más útiles que otras. En concreto, mientras que algunas formulaciones son especialmente adecuadas a la hora de evaluar numéricamente la *FVE* en un punto del espacio paramétrico, otras lo son para interpretar las propiedades del proceso ARMA cuyos parámetros se desea estimar, así como para diagnosticar dicho modelo y para obtener previsiones a partir de él. No obstante, como se muestra en este capítulo, es posible encontrar una formulación adecuada para todos estos propósitos. A continuación se resume brevemente el contenido de este capítulo:

Sección 2.1: Formulación de los procesos ARMA estacionarios. En esta sección introductoria se describe la formulación matemática y estadística de los modelos objeto de estudio, así como sus condiciones de estacionariedad e invertibilidad. Estos temas constituyen la base para la obtención de la *FVE* de dichos modelos.

Sección 2.2: Repaso de procedimientos existentes. En esta sección se revisan los procedimientos más relevantes disponibles actualmente para evaluar la *FVE*. Aunque el contenido de esta sección es, básicamente, un repaso de temas conocidos, también se presentan en detalle ciertas cuestiones computacionalmente interesantes, ignoradas habitualmente en los trabajos teóricos sobre el tema.

Sección 2.3: Un nuevo procedimiento para evaluar la función de verosimilitud exacta. A partir de algunas ideas sugeridas por los procedimientos descritos en la sección anterior, se puede diseñar un método computacionalmente eficiente para evaluar la *FVE*, que tiene ciertas ventajas respecto a los procedimientos disponibles en la actualidad. Además, este método permite demostrar la equivalencia matemática (aunque no computacional) de las formulaciones más utilizadas entre las propuestas en la literatura y puede utilizarse como punto de partida para el diseño de otros métodos de estimación, alternativos al de máxima verosimilitud exacta, que, en la práctica, pueden resultar extremadamente útiles

(*vid.* Capítulo 3). En esta sección se describe con todo detalle este nuevo procedimiento, prestando especial atención a determinados aspectos computacionales y numéricos, que resultan fundamentales para su puesta en práctica.

Sección 2.4: Análisis comparativo de distintos procedimientos. Finalmente, en esta sección se comparan la eficiencia computacional (en términos de tiempo de cálculo, precisión numérica y necesidades de almacenamiento) y las ventajas prácticas de los métodos descritos en las secciones 2.2 y 2.3. El procedimiento de la sección 2.3 resulta al menos tan eficiente como el mejor de los disponibles actualmente, con el añadido de presentar ciertas ventajas prácticas sobre estos últimos.

2.1. Formulación de los procesos ARMA estacionarios

A lo largo del presente trabajo, supondremos que se dispone de N observaciones o realizaciones simultáneas de m variables aleatorias ordenadas en el tiempo. Denotaremos por $\mathbf{y}_t = (y_{1t}, y_{2t}, \dots, y_{mt})^T$ al vector ($m \times 1$) del nivel de cada serie original en el período t ($t = 1, 2, \dots, N$). Supondremos también que cada una de las m series consideradas admite una transformación de Box-Cox, caracterizada por dos parámetros (λ_i, m_i) y definida como:

$$y_{it}^{(\lambda_i, m_i)} = \begin{cases} \frac{(y_{it} + m_i)^{\lambda_i} - 1}{\lambda_i} & , \text{ si } 0 < |\lambda_i| \leq 2 \\ \ln(y_{it} + m_i) & , \text{ si } \lambda_i = 0 \end{cases} \quad [2.1.1]$$

que estabiliza su varianza e induce linealidad y normalidad [*vid.* Box y Cox (1964); Box y Jenkins (1970), cap. 4; Jenkins y Alavi (1981)]. Por otro lado, si ξ_{it} es el componente determinista de la i -ésima serie transformada en el período t , definimos el componente puramente estocástico de dicha serie como:

$$z_{it} = y_{it}^{(\lambda_i, m_i)} - \xi_{it} \quad [2.1.2]$$

Por tanto, el vector ($m \times 1$) $\mathbf{z}_t = (z_{1t}, z_{2t}, \dots, z_{mt})^T$ es el vector del nivel de los componentes puramente estocásticos de las series transformadas en el período t ($t = 1, 2, \dots, N$). Finalmente, definimos $\mathbf{w}_t = \mathbf{V}(\mathbf{B})\mathbf{z}_t$, con $\mathbf{V}(\mathbf{B}) = \text{diag}[V_1(\mathbf{B}), V_2(\mathbf{B}), \dots, V_m(\mathbf{B})]$, de manera que cada serie $w_{it} = V_i(\mathbf{B})z_{it}$ ($i = 1, 2, \dots, m$) resultante de esta transformación sea estacionaria en media y en covarianza¹. La matriz ($m \times m$) diagonal $\mathbf{V}(\mathbf{B})$ tiene como elemento característico un polinomio $V_i(\mathbf{B})$ definido en el operador de retardos \mathbf{B} ($\mathbf{B}^k \mathbf{z}_t = \mathbf{z}_{t-k}$),

normalizado ($V_i(0)=1$) y de orden d_i , que contiene los factores no estacionarios del modelo ($V_i(B)=0 \Rightarrow |B|=1$) (vid. Jenkins y Alavi (1981); Tiao y Box (1981); Wei (1990), cap. 14).

2.1.1. Procesos ARMA(p,q) multivariantes estocásticos

En estas condiciones, diremos que la *serie temporal múltiple* w_t ($t=1,2,\dots,n$), con $n=N-\text{máximo}\{d_i, i=1,2,\dots,m\}$, ha sido generada por un *proceso ARMA(p,q) multivariante estocástico estacionario* (vid. Hannan (1970); Zellner y Palm (1974); Wallis (1977); Jenkins y Alavi (1981); Tiao y Box (1981); Granger y Newbold (1986), cap. 8; Mills (1990), cap. 14; Wei (1990), cap. 14) cuando dicha serie admita la siguiente representación:

$$\Phi(B)(w_t - \mu) = \Theta(B)a_t, \quad t=1,2,\dots,n \quad [2.1.3]$$

donde:

$$\Phi(B) = I_m - \Phi_1 B - \Phi_2 B^2 - \dots - \Phi_p B^p \quad [2.1.4]$$

$$\Theta(B) = I_m - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q \quad [2.1.5]$$

y μ es el vector que contiene la media de cada serie. En [2.1.4] y [2.1.5], Φ_i ($i=1,2,\dots,p$) y θ_i ($i=1,2,\dots,q$) son matrices ($m \times m$) de parámetros, I_m es la matriz identidad de orden m y B es el operador de retardos. En principio, supondremos que los polinomios de [2.1.4] y [2.1.5] satisfacen las siguientes condiciones:

$$|\Phi(B)| = 0 \Rightarrow |B| > 1 \quad [2.1.6]$$

$$|\Theta(B)| = 0 \Rightarrow |B| > 1 \quad [2.1.7]$$

a las que nos referiremos como condiciones de *estacionariedad* y de *invertibilidad*, respectivamente (vid. Jenkins y Alavi (1981); Tiao y Box (1981); Wei (1990), cap. 14). Dichas condiciones aseguran que las siguientes sumas infinitas:

$$\Psi(B) = \Phi(B)^{-1}\Theta(B) = I_m + \sum_{j=1}^{\infty} \Psi_j B^j \quad [2.1.8]$$

$$\Pi(B) = \Theta(B)^{-1}\Phi(B) = I_m - \sum_{j=1}^{\infty} \Pi_j B^j \quad [2.1.9]$$

convergen para $|B| \leq 1$. En concreto, [2.1.6] y [2.1.7] aseguran que las secuencias de matrices Ψ_j y Π_j se amortiguan a medida que el índice j crece, de manera que, por un lado, las matrices de covarianzas de w_t son finitas y, por otro lado, w_t depende menos de su pasado más lejano que de su pasado reciente.

Por último, en [2.1.3], supondremos que a_t es una sucesión de vectores $(m \times 1)$ aleatorios *idéntica e independientemente distribuidos* caracterizados por:

$$E[a_t] = \mathbf{0}_{(m \times 1)} \quad [2.1.10]$$

$$E[a_t a_{t+k}^T] = \begin{cases} \sigma^2 \mathbf{Q}_{(m \times m)}, & k=0 \\ \mathbf{0}_{(m \times m)}, & k \neq 0 \end{cases} \quad [2.1.11]$$

con $\sigma^2 > 0$ y \mathbf{Q} simétrica y definida positiva². Por tanto, las *matrices de covarianzas teóricas* del proceso estacionario w_t están dadas por:

$$E[\tilde{w}_t \tilde{w}_{t+k}^T] = E[\tilde{w}_{t-k} \tilde{w}_t^T] = \sigma^2 \Gamma(k) = \sigma^2 \sum_{j=0}^{\infty} \Psi_j \mathbf{Q} \Psi_{j+k}^T = \sigma^2 \Gamma(-k)^T, \quad k=0,1,2,\dots \quad [2.1.12]$$

donde $\tilde{w}_t = w_t - \mu$. Las matrices Ψ_j pueden calcularse recursivamente de la siguiente manera:

$$\Psi_j = \begin{cases} \mathbf{I}_m, & j=0 \\ \sum_{i=1}^j \Phi_i \Psi_{j-i} - \Theta_j, & j \geq 1 \end{cases} \quad [2.1.13]$$

y se entiende que $\Phi_j = \mathbf{0}$ para $j > p$ y $\Theta_j = \mathbf{0}$ para $j > q$. Una recursión similar puede utilizarse para calcular las matrices Π_j de [2.1.9]:

$$\Pi_j = \begin{cases} \mathbf{I}_m, & j=0 \\ \Phi_j + \sum_{i=1}^j \Theta_i \Pi_{j-i}, & j \geq 1 \end{cases} \quad [2.1.14]$$

Alternativamente [vid. [2.2.8] y (3.7) de Jenkins y Alavi (1981)], las matrices de covarianzas teóricas pueden expresarse como:

$$\sigma^2 \mathbf{\Gamma}(k) = \sigma^2 \sum_{i=1}^p \mathbf{\Gamma}(k-i) \mathbf{\Phi}_i^T + \sigma^2 \mathbf{\Gamma}_{wa}(k) - \sigma^2 \sum_{j=1}^q \mathbf{\Gamma}_{wa}(k-j) \mathbf{\Theta}_j^T, \quad k=0,1,2,\dots \quad [2.1.15]$$

donde $\sigma^2 \mathbf{\Gamma}_{wa}(h) = E[\tilde{\mathbf{w}}_t \mathbf{a}_{t+h}^T]$ son las matrices de covarianzas cruzadas entre \mathbf{w}_t y \mathbf{a}_t (vid. [2.2.4]-[2.2.6]).

2.1.2. Formulación compacta de procesos ARMA(p,q) multivariantes estocásticos

Para obtener una forma cerrada de la FVE de las observaciones \mathbf{w}_t generadas por el proceso [2.1.3], resulta conveniente escribir [2.1.3] para $t=1,2,\dots,n$ y reorganizar el resultado de forma compacta. Para ello [vid. Hillmer y Tiao (1979); Ljung y Box (1979); Nicholls y Hall (1979)], definimos los vectores de observaciones $\tilde{\mathbf{w}}$ ($nm \times 1$), de residuos \mathbf{a} ($nm \times 1$) y de valores premuestrales \mathbf{u}_* ($(p+q)m \times 1$) como:

$$\tilde{\mathbf{w}} = \begin{bmatrix} \tilde{\mathbf{w}}_1 \\ \tilde{\mathbf{w}}_2 \\ \vdots \\ \tilde{\mathbf{w}}_n \end{bmatrix}; \quad \mathbf{a} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_n \end{bmatrix}; \quad \mathbf{u}_* = \begin{bmatrix} \tilde{\mathbf{w}}_{1-p} \\ \tilde{\mathbf{w}}_{2-p} \\ \vdots \\ \tilde{\mathbf{w}}_0 \\ \dots\dots\dots \\ \mathbf{a}_{1-q} \\ \mathbf{a}_{2-q} \\ \vdots \\ \mathbf{a}_0 \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{w}}_* \\ \dots\dots\dots \\ \mathbf{a}_* \end{bmatrix} \quad [2.1.16]$$

donde $\tilde{\mathbf{w}}_*$ es $(pm \times 1)$ y \mathbf{a}_* es $(qm \times 1)$. Según esto, el proceso [2.1.3] puede escribirse para $t=1,2,\dots,n$ de forma compacta como sigue:

$$\mathbf{D}_{\Phi,n} \tilde{\mathbf{w}} = \mathbf{D}_{\Theta,n} \mathbf{a} + \mathbf{V} \mathbf{u}_*$$

[2.1.17]

donde \mathbf{V} es una matriz de orden $(nm \times (p+q)m)$, definida en dos bloques como:

$$\mathbf{V} = \begin{pmatrix} \mathbf{G}_{\Phi,n} & \mathbf{G}_{\Theta,n} \end{pmatrix} \quad [2.1.18]$$

y las matrices $\mathbf{D}_{\Phi,n}$ ($mn \times mn$), $\mathbf{D}_{\Theta,n}$ ($mn \times mn$), $\mathbf{G}_{\Phi,n}$ ($mn \times mp$) y $\mathbf{G}_{\Theta,n}$ ($mn \times mq$) se definen como:

$$G_{\boldsymbol{\theta},n} = \begin{bmatrix} -\boldsymbol{\theta}_q & -\boldsymbol{\theta}_{q-1} & -\boldsymbol{\theta}_{q-2} & \cdots & -\boldsymbol{\theta}_1 \\ \mathbf{0} & -\boldsymbol{\theta}_q & -\boldsymbol{\theta}_{q-1} & \cdots & -\boldsymbol{\theta}_2 \\ \mathbf{0} & \mathbf{0} & -\boldsymbol{\theta}_q & \cdots & -\boldsymbol{\theta}_3 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & -\boldsymbol{\theta}_q \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \quad [2.1.22]$$

A la vista de [2.1.20], resulta inmediato comprobar [vid. Hillmer y Tiao (1979)] que:

$$D_{\boldsymbol{\theta},n}^{-1} = \begin{bmatrix} \mathbf{I}_m & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{E}_1 & \mathbf{I}_m & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{E}_2 & \mathbf{E}_1 & \mathbf{I}_m & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \mathbf{E}_{n-1} & \cdots & \mathbf{E}_2 & \mathbf{E}_1 & \mathbf{I}_m \end{bmatrix} \quad [2.1.23]$$

donde las n matrices ($m \times m$) \mathbf{E}_k ($k=1,2,\dots,n-1$) pueden calcularse recursivamente de la siguiente forma:

$$\mathbf{E}_k = \sum_{j=1}^q \boldsymbol{\theta}_j \mathbf{E}_{k-j}, \quad k=1,2,\dots,n-1 \quad [2.1.24]$$

con $\mathbf{E}_0 = \mathbf{I}_m$ y $\mathbf{E}_k = \mathbf{0}$ para $k < 0$.

Para concluir, nótese, en primer lugar, que la representación [2.1.17] para un proceso AR(p) puro ($q=0$) se reduce a $D_{\boldsymbol{\Phi},n} \tilde{\mathbf{w}} = \mathbf{a} + V\mathbf{u}_*$, con $V = G_{\boldsymbol{\Phi},n}$ y $\mathbf{u}_* = \tilde{\mathbf{w}}_*$; análogamente, para un MA(q) puro ($p=0$), tenemos $\tilde{\mathbf{w}} = D_{\boldsymbol{\theta},n} \mathbf{a} + V\mathbf{u}_*$, con $V = G_{\boldsymbol{\theta},n}$ y $\mathbf{u}_* = \mathbf{a}_*$. Y, en segundo lugar, a la vista de [2.1.11], resulta evidente que:

$$E[\mathbf{a}\mathbf{a}^T] = \sigma^2(\mathbf{I}_n \otimes \mathbf{Q}) \quad [2.1.25]$$

donde \otimes representa el *producto de Kronecker* (vid. Apéndice A.1). Si suponemos que $\mathbf{a}_t \sim i.i.d. N(\mathbf{0}, \sigma^2 \mathbf{Q})$ para todo t , el proceso ARMA multivariante estocástico queda completamente especificado, de manera que ya se puede formular la *FVE* de la muestra considerada.

2.1.3. Referencias y notas para la sección 2.1

Tratamientos sencillos y actuales de la especificación y propiedades de procesos ARMA multivariantes, pueden encontrarse en Granger y Newbold (1986), cap. 8, Mills (1990), cap. 14, y Wei (1990), cap. 14. Estos libros, además, constituyen una fuente considerable de referencias bibliográficas. Concretamente, en Hannan (1970), Zellner y Palm (1974) y Wallis (1977), se discuten, entre otros temas, la unicidad de la representación [2.1.3] y su relación con los modelos de ecuaciones simultáneas tradicionales. En Jenkins y Alavi (1981) puede encontrarse un resumen de la interpretación e implicaciones de estos modelos, incluyendo las condiciones [2.1.6] y [2.1.7] y la posible incorporación de estructura estacional. Este último tema también es considerado por Tiao y Box (1981), quienes, además, generalizan la condición [2.1.6] permitiendo que algunas raíces de la ecuación característica $|\Phi(\mathbf{B})| = 0$ estén sobre el círculo unitario, con el fin de evitar la no invertibilidad provocada por la sobre-diferenciación de las series estocásticas transformadas y los problemas derivados de ello. Además, Jenkins y Alavi (1981), Tiao y Box (1981) y Tiao y Tsay (1983, 1989) son puntos de referencia fundamentales sobre los principios de construcción de modelos ARMA multivariantes.

[1] Que *cada* serie sea estacionaria, junto con las condiciones [2.1.6] y [2.1.7], *no* es la única forma de especificar un modelo ARMA multivariante estocástico. Para ganar generalidad, podrían incluirse en el vector \mathbf{w} variables no estacionarias siempre y cuando estuviesen *cointegradas*. En el contexto de un modelo ARMA multivariante estocástico, la cointegración podría reflejarse en la existencia de raíces del polinomio $|\Phi(\mathbf{B})| = 0$ con módulo $|\mathbf{B}| = 1$ sin que existiesen factores comunes con módulo unitario en ninguna columna de la matriz $\Phi(\mathbf{B})$. No obstante, como el objeto de este trabajo es tan sólo la estimación de este tipo de modelos, se da por supuesto que estas circunstancias y cualesquiera otras relacionadas con la modelización de series multivariantes estocásticas ya han sido tenidas o se tendrán posteriormente en cuenta.

[2] La descomposición de la matriz de covarianzas [2.1.11] es un artificio empleado simplemente por conveniencia computacional. Brevemente, dicha descomposición permite *concentrar* la *FVE* en el estimador maximoverosímil de σ^2 , con lo que (i) se consigue una aceptable estabilidad numérica en los cálculos, al tener el resto de parámetros a estimar un orden de magnitud homogéneo y (ii) el logaritmo de la *FVE* concentrada puede escribirse de una forma muy adecuada para su maximización mediante algoritmos eficientes de propósito general

(métodos *cuasi-Newton*) y algoritmos específicos para sumas de cuadrados (tipo *Gauss-Newton*). Todos estos temas son tratados con detalle en el Capítulo 3.

2.2. Repaso de procedimientos existentes

La utilidad inmediata de la *FVE* asociada a un vector de series temporales generado por [2.1.3], reside en la estimación de los parámetros que aparecen en el modelo ($\Phi_i, i=1,2,\dots,p, \Theta_i, i=1,2,\dots,q, \mu, \sigma^2$ y Q), tarea que se llevará a cabo maximizando la función obtenida (o alguna aproximación a ella; *vid.* Capítulo 3). Además, el estudio de la forma funcional de la *FVE* y de las distintas expresiones de sus componentes, resulta fundamental para entender tanto sus propiedades estructurales como las del proceso ARMA cuyos parámetros se desea estimar.

A partir de la exposición de la metodología Box-Jenkins para el análisis univariante de series temporales [Box y Jenkins (1970)], la fase de la estimación de los parámetros del modelo previamente identificado ha recibido una considerable atención en la literatura. Por ello, puede resultar útil presentar un breve resumen de los resultados más relevantes conocidos hasta hoy.

En el dominio del tiempo, Newbold (1974), Ali (1977), Dent (1977) y Ljung y Box (1979) obtienen expresiones alternativas de la *FVE* de procesos univariantes; los dos últimos trabajos también describen con cierto detalle cómo evaluar numéricamente sus expresiones. A partir de Ansley (1979), el interés sobre el tema se desvía hacia la evaluación y la maximización práctica de la *FVE*, en vez de hacia la obtención de expresiones analíticas calculables de la *FVE*. Estos trabajos se basan siempre en la aplicación de técnicas de modelización en *espacio de los estados* [*vid.* Cancio (1989)] y culminan en Kohn y Ansley (1985), donde también se obtienen expresiones de las primeras y segundas derivadas de la *FVE* de procesos univariantes. En Kohn y Ansley (1985) se combinan los algoritmos de Ansley (1979) y Pearlman (1980) [programado en Melard (1984)] para evaluar la *FVE* de procesos univariantes de la forma más eficiente, desde el punto de vista computacional, conocida en la literatura [el método de Kohn y Ansley (1985) es superior, en concreto, al basado en el *filtro de Kalman* de Gardner, Harvey y Phillips (1980) y al basado en las *ecuaciones de Chandrasekhar* de Pearlman (1980)]. Como contrapartida a su eficiencia, el método de Kohn y Ansley (1985) es de una complejidad extrema, debido a lo cual la interpretación de las propiedades estructurales de la *FVE* resulta difícil.

La literatura sobre estimación de modelos univariantes en el dominio del tiempo, se completa con los trabajos de McLeod (1975, 1977) y Tunnicliffe Wilson (1979) (sobre el cálculo eficiente de las autocovarianzas teóricas, necesarias para evaluar la *FVE*), Dent y Min (1978) y Ansley y Newbold (1980) (donde se comparan las propiedades en muestras pequeñas de los distintos estimadores propuestos en la literatura; *vid.* Capítulo 3) y Godolphin y Unwin (1983) (un procedimiento sencillo de calcular la matriz de covarianzas del estimador por máxima verosimilitud). Finalmente, Tuan (1987) describe una forma de evaluar la *FVE* y sus primeras derivadas haciendo uso de la técnica de la *retrovisión* de Box y Jenkins (1970) y Koreisha y Pukkila (1990) presentan un estimador por mínimos cuadrados generalizados que equivale numéricamente, bajo ciertas condiciones, al estimador por máxima verosimilitud.

Las técnicas basadas en la representación de procesos ARMA en espacio de los estados, han sido aún más utilizadas recientemente en la literatura sobre estimación de procesos ARMA multivariantes (*vid.* apartado 2.2.3). De hecho, desde los trabajos de Hillmer y Tiao (1979), Nicholls y Hall (1979) y Hall y Nicholls (1980), ninguna referencia en la literatura (hasta donde conocemos) se ha dedicado a la obtención, evaluación y maximización práctica de la *FVE* de procesos ARMA multivariantes en el dominio del tiempo como un problema en sí mismo en vez de como un caso particular de la estimación de modelos más generales, ya que puede demostrarse que la aplicación de las técnicas basadas en la representación de procesos ARMA en espacio de los estados, no es sino un caso particular de esa misma estrategia aplicada a la representación general de cualquier modelo lineal y dinámico en forma espacio de los estados [*vid.* Cancio (1989) y Sotoca (1992)].

En Shea (1989) se presenta un algoritmo para evaluar la *FVE*, aplicando el *filtro de Chandrasekhar* [*vid.* Morf, Sidhu y Kailath (1974); Shea (1987)] a la representación en espacio de los estados del proceso ARMA multivariante cuyos parámetros se desea estimar. Este método es más eficiente que el de Ansley y Kohn (1983) [que es una generalización de Jones (1980), basado en el *filtro de Kalman*; *vid.* Kalman (1960) y Anderson y Moore (1979)] cuando no faltan observaciones en la muestra considerada.

Sin hacer uso de estas técnicas, el trabajo de Tunnicliffe Wilson (1973) puede considerarse pionero en el contexto de la estimación de procesos ARMA multivariantes. Posteriormente, Osborn (1977), Phadke y Kedem (1978) y Hillmer y Tiao (1979) han obtenido expresiones de la *FVE* para un proceso MA puro. En este último trabajo también se propone una aproximación a la *FVE* de procesos mixtos. Estrictamente hablando, sólo Nicholls y Hall (1979) [con el énfasis puesto en la práctica en Hall y Nicholls (1980)] han derivado una expresión *operativa* de la *FVE* para procesos ARMA multivariantes en el dominio del tiempo sin recurrir a las técnicas basadas en la representación en espacio de los estados.

Finalmente, Koreisha y Pukkila (1989) discuten varios procedimientos para obtener estimaciones preliminares de los parámetros, mientras que Ansley (1980) presenta un procedimiento eficiente [mejorado en Kohn y Ansley (1982)] para calcular las matrices de autocovarianzas teóricas, necesarias para evaluar la *FVE*.

A la vista de todos estos resultados, nuestros objetivos en este capítulo son: [1] exponer y comparar los distintos procedimientos disponibles actualmente para evaluar la *FVE* (secciones 2.2 y 2.4) y [2] describir en detalle un nuevo procedimiento para llevar a cabo dicha evaluación (sección 2.3). Este procedimiento permite [3] demostrar la equivalencia matemática entre las formulaciones más relevantes propuestas en la literatura (sección 2.3) y [4] interpretar de distintas formas los términos que aparecen en la expresión de la *FVE*, así como generar los medios necesarios para diagnosticar el modelo ARMA estimado y para obtener previsiones a partir de él (sección 2.3). Es importante subrayar que algunas de estas últimas cuestiones no son inmediatas cuando se hace uso de las técnicas basadas en la representación en espacio de los estados [por ejemplo, Kohn y Ansley (1985); Shea (1989)]. Además, a partir del procedimiento de la sección 2.3, pueden diseñarse otros métodos de estimación, alternativos al de máxima verosimilitud exacta que, en la práctica, pueden resultar muy útiles (*vid.* Capítulo 3).

2.2.1. Evaluación directa de la función de verosimilitud exacta

La *FVE* de los parámetros Φ_i ($i=1,2,\dots,p$), Θ_i ($i=1,2,\dots,q$), μ , σ^2 y \mathbf{Q} , para una muestra \mathbf{w} dada, puede escribirse, bajo la hipótesis $\mathbf{a} \sim N[\mathbf{0}, \sigma^2(\mathbf{I}_n \otimes \mathbf{Q})]$, como:

$$L(\Phi_i, \Theta_i, \mu, \sigma^2, \mathbf{Q} | \mathbf{w}) = (2\pi\sigma^2)^{-\frac{nm}{2}} \cdot |\Sigma|^{-\frac{1}{2}} \cdot \exp \left[-\frac{1}{2\sigma^2} \tilde{\mathbf{w}}^T \Sigma^{-1} \tilde{\mathbf{w}} \right]$$

[2.2.1]

donde $\Sigma = (1/\sigma^2)E[\tilde{\mathbf{w}}\tilde{\mathbf{w}}^T]$ es la matriz ($nm \times nm$) de autocovarianzas teóricas del proceso w_t generado por [2.1.3]. Es inmediato comprobar que $\Sigma_{ij} = (1/\sigma^2)E[\tilde{w}_i\tilde{w}_j^T] = \Gamma(j-i)$ y que $\Sigma_{ji} = (1/\sigma^2)E[\tilde{w}_j\tilde{w}_i^T] = \Gamma(i-j) = \Gamma(j-i)^T = \Sigma_{ij}^T$ ($i,j=1,2,\dots,n$); luego para obtener la matriz Σ necesitamos calcular $\Gamma(k)$ desde $k=0$ hasta $k=n-1$ (n matrices ($m \times m$) distintas). Para calcular estas n matrices, conviene expresar el proceso multivariante [2.1.3] en términos de [2.1.4] y [2.1.5]:

$$\tilde{w}_t - \Phi_1 \tilde{w}_{t-1} - \Phi_2 \tilde{w}_{t-2} - \cdots - \Phi_p \tilde{w}_{t-p} = a_t - \Theta_1 a_{t-1} - \Theta_2 a_{t-2} - \cdots - \Theta_q a_{t-q} \quad [2.2.2]$$

Entonces, postmultiplicando [2.2.2] por a_{t+k}^T , tomando valores esperados y reordenando la expresión resultante, tenemos que:

$$\begin{aligned} E[\tilde{w}_t a_{t+k}^T] &= \Phi_1 E[\tilde{w}_{t-1} a_{t+k}^T] + \Phi_2 E[\tilde{w}_{t-2} a_{t+k}^T] + \cdots + \Phi_p E[\tilde{w}_{t-p} a_{t+k}^T] + \\ &E[a_t a_{t+k}^T] - \Theta_1 E[a_{t-1} a_{t+k}^T] - \Theta_2 E[a_{t-2} a_{t+k}^T] - \cdots - \Theta_q E[a_{t-q} a_{t+k}^T] \end{aligned} \quad [2.2.3]$$

Si tenemos en cuenta que $E[a_{t+j} a_t^T] = \mathbf{0}$ para $j \neq 0$ (a_t es ruido blanco) y que $E[\tilde{w}_{t-j} a_t^T] = \mathbf{0}$ para $j > 0$ (\tilde{w}_t no depende de valores futuros de a_t), resulta inmediato comprobar los siguientes resultados:

$$\Gamma_{wa}(0) = \frac{1}{\sigma^2} E[\tilde{w}_t a_t^T] = \mathbf{Q} \quad [2.2.4]$$

$$\Gamma_{wa}(k) = \frac{1}{\sigma^2} E[\tilde{w}_t a_{t+k}^T] = \mathbf{0}, \quad k > 0 \quad [2.2.5]$$

$$\Gamma_{wa}(-k) = \frac{1}{\sigma^2} E[\tilde{w}_t a_{t-k}^T] = \sum_{i=1}^k \Phi_i \Gamma_{wa}(-k+i) - \Theta_k \mathbf{Q}, \quad k=1,2,\dots \quad [2.2.6]$$

En esta última expresión, $\Phi_i = \mathbf{0}$ para $i > p$ y $\Theta_i = \mathbf{0}$ para $i > q$. Por tanto, dada \mathbf{Q} , las matrices $\Gamma_{wa}(k)$ para $k < 0$ pueden calcularse recursivamente según la expresión [2.2.6]. Por otro lado, para obtener los elementos de la matriz Σ , transponemos [2.2.2], premultiplicamos el resultado por \tilde{w}_{t-k} y tomamos valores esperados, de forma que, reordenando la expresión resultante:

$$\begin{aligned} E[\tilde{w}_{t-k} \tilde{w}_t^T] &= E[\tilde{w}_{t-k} \tilde{w}_{t-1}^T] \Phi_1^T + E[\tilde{w}_{t-k} \tilde{w}_{t-2}^T] \Phi_2^T + \cdots + E[\tilde{w}_{t-k} \tilde{w}_{t-p}^T] \Phi_p^T + \\ &E[\tilde{w}_{t-k} a_t^T] - E[\tilde{w}_{t-k} a_{t-1}^T] \Theta_1^T - E[\tilde{w}_{t-k} a_{t-2}^T] \Theta_2^T - \cdots - E[\tilde{w}_{t-k} a_{t-q}^T] \Theta_q^T \end{aligned} \quad [2.2.7]$$

Escribiendo [2.2.7] para $k=0,1,2,\dots$ y dividiendo por σ^2 , se obtiene lo siguiente (*vid.* [2.1.15]):

$$\Gamma(k) = \sum_{i=1}^p \Gamma(k-i) \Phi_i^T + \Gamma_{wa}(k) - \sum_{j=1}^q \Gamma_{wa}(k-j) \Theta_j^T, \quad k=0,1,2,\dots \quad [2.2.8]$$

Recordando que $\Gamma(h)^T = \Gamma(-h)$, [2.2.8] puede escribirse para $k=0,1,2,\dots,p$, sin más que transponer dicha expresión, como sigue:

$$\Gamma(-k) - \sum_{i=1}^p \Phi_i \Gamma(i-k) = \mathbf{W}(k) , k=0,1,2,\dots,p \quad [2.2.9]$$

donde:

$$\mathbf{W}(k) = \Gamma_{wa}(k)^T - \sum_{j=1}^q \Theta_j \Gamma_{wa}(k-j)^T , k=0,1,2,\dots,p \quad [2.2.10]$$

Definiendo $\gamma(h) = \text{vec}[\Gamma(h)]$ y $\omega(h) = \text{vec}[\mathbf{W}(h)]$ y aplicando el operador vec (vid. Apéndice A.1) a [2.2.9], tenemos que:

$$\gamma(-k) - \sum_{i=1}^p (\mathbf{I}_m \otimes \Phi_i) \gamma(i-k) = \omega(k) , k=0,1,\dots,p \quad [2.2.11]$$

En esta expresión, aparecen como incógnitas $\gamma(-p), \gamma(-p+1), \dots, \gamma(-1), \gamma(0), \gamma(1), \dots, \gamma(p-1)$ y $\gamma(p)$. Pero como $\gamma(-h) = \text{vec}[\Gamma(-h)] = \text{vec}[\Gamma(h)^T] = \mathbf{P} \text{vec}[\Gamma(h)] = \mathbf{P} \gamma(h)$, para alguna *matriz de permutación* \mathbf{P} ($m^2 \times m^2$) (vid. Apéndice A.1), [2.2.11] puede expresarse sólo en términos de $\gamma(0), \gamma(1), \dots, \gamma(p-1)$ y $\gamma(p)$ como:

$$\mathbf{P} \gamma(k) - \sum_{i=1}^k (\mathbf{I}_m \otimes \Phi_i) \mathbf{P} \gamma(k-i) - \sum_{i=k+1}^p (\mathbf{I}_m \otimes \Phi_i) \gamma(i-k) = \omega(k) , k=0,1,\dots,p \quad [2.2.12]$$

Esta expresión constituye un sistema lineal de $m^2(p+1)$ ecuaciones con $m^2(p+1)$ incógnitas [los m^2 elementos de cada uno de los $p+1$ vectores $\gamma(0), \gamma(1), \dots, \gamma(p)$]. Dicho sistema puede escribirse de forma compacta como $\Phi \gamma = \omega$, donde $\gamma^T = [\gamma(0)^T, \gamma(1)^T, \dots, \gamma(p)^T]$ ($m^2(p+1) \times 1$), $\omega^T = [\omega(0)^T, \omega(1)^T, \dots, \omega(p)^T]$ ($m^2(p+1) \times 1$) y la matriz Φ ($m^2(p+1) \times m^2(p+1)$) está definida como $\Phi = \mathbf{F}_1 + \mathbf{F}_2$, donde:

$$\mathbf{F}_1 = \begin{pmatrix} \mathbf{I}_{m^2} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ -(\mathbf{I}_m \otimes \Phi_1) & \mathbf{P} & \mathbf{0} & \dots & \mathbf{0} & \mathbf{0} \\ -(\mathbf{I}_m \otimes \Phi_2) & -(\mathbf{I}_m \otimes \Phi_1) \mathbf{P} & \mathbf{P} & \dots & \mathbf{0} & \mathbf{0} \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ -(\mathbf{I}_m \otimes \Phi_{p-1}) & -(\mathbf{I}_m \otimes \Phi_{p-2}) \mathbf{P} & -(\mathbf{I}_m \otimes \Phi_{p-3}) \mathbf{P} & \dots & \mathbf{P} & \mathbf{0} \\ -(\mathbf{I}_m \otimes \Phi_p) & -(\mathbf{I}_m \otimes \Phi_{p-1}) \mathbf{P} & -(\mathbf{I}_m \otimes \Phi_{p-2}) \mathbf{P} & \dots & -(\mathbf{I}_m \otimes \Phi_1) \mathbf{P} & \mathbf{P} \end{pmatrix} \quad [2.2.13]$$

$$F_2 = \begin{pmatrix} \mathbf{0} & -(\mathbf{I}_m \otimes \Phi_1) & -(\mathbf{I}_m \otimes \Phi_2) & \cdots & -(\mathbf{I}_m \otimes \Phi_{p-1}) & -(\mathbf{I}_m \otimes \Phi_p) \\ \mathbf{0} & -(\mathbf{I}_m \otimes \Phi_2) & -(\mathbf{I}_m \otimes \Phi_3) & \cdots & -(\mathbf{I}_m \otimes \Phi_p) & \mathbf{0} \\ \mathbf{0} & -(\mathbf{I}_m \otimes \Phi_3) & -(\mathbf{I}_m \otimes \Phi_4) & \cdots & \mathbf{0} & \mathbf{0} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \mathbf{0} & -(\mathbf{I}_m \otimes \Phi_p) & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{pmatrix} \quad [2.2.14]$$

La primera columna de la matriz F_1 se ha escrito teniendo en cuenta que $\Gamma(0)$ es una matriz simétrica, por lo que $P\gamma(0)=\gamma(0)$ y entonces no es necesario postmultiplicar ningún componente de la primera columna de F_1 por P . Una vez resuelto el sistema [2.2.12] y recuperadas las matrices $\Gamma(k)$, $k=0,1,\dots,p$, del vector γ , las restantes matrices $\Gamma(k)$, $k=p+1,p+2,\dots,n-1$, pueden calcularse recursivamente según [2.2.8]. El procedimiento CGAMMA del Apéndice A.2 puede utilizarse para llevar a cabo todos estos cálculos. Un procedimiento más eficiente, aunque bastante más complicado, para calcular las matrices de autocovarianzas teóricas, puede encontrarse en Kohn y Ansley (1982). Finalmente, es interesante señalar que cuando $m=1$, el mecanismo descrito [que es equivalente al de Hall y Nicholls (1980)] se reduce al propuesto por Ljung y Box (1979).

Una vez obtenida la matriz de autocovarianzas Σ , la evaluación de la FVE [2.2.1] resulta, al menos en teoría, inmediata. No obstante, en la práctica, la utilización *directa* de [2.2.1] para evaluar la FVE puede resultar ineficiente, en términos tanto de tiempo de cálculo como de necesidades de almacenamiento. Por esta razón, toda la literatura sobre la forma de evaluar la FVE se reduce a la obtención de expresiones *operativas* del determinante $|\Sigma|$ y de la forma cuadrática $\tilde{w}^T \Sigma^{-1} \tilde{w}$ que aparecen en [2.2.1].

A pesar de estas circunstancias, la expresión [2.2.1] es interesante para ciertos propósitos. En primer lugar, permite contrastar la validez numérica de otros procedimientos más eficientes para evaluar la FVE, ya que la codificación de un procedimiento basado en el cálculo directo de $|\Sigma|$ y de $\tilde{w}^T \Sigma^{-1} \tilde{w}$ es muy sencilla (como se discute más abajo). En segundo lugar, la comparación de los componentes de [2.2.1] con los resultantes de otros procedimientos más eficientes, permite interpretar dichos componentes en términos de las características estructurales del proceso generador de la muestra. Por último, cuando $p=0$ (es decir, cuando el proceso es un MA(q) puro), la evaluación directa de [2.2.1] sí puede resultar eficiente en la práctica, ya que, en tal caso, $\Gamma(k)=\mathbf{0}$ para $k>q$ y entonces Σ es un *matriz simétrica de*

banda con anchura de banda máxima igual a mq [esta última circunstancia es, de hecho, la que se encuentra detrás del procedimiento de Ansley (1979)].

La forma más eficiente y numéricamente estable de evaluar directamente $|\Sigma|$ y $\tilde{w}^T \Sigma^{-1} \tilde{w}$, consiste en obtener la *descomposición de Cholesky* [vid. Martin, Peters y Wilkinson (1965)] de la matriz $\Sigma = PP^T$, donde P es una matriz $(nm \times nm)$ triangular inferior. Entonces, $|\Sigma| = |P|^2$ ($|P|$ es simplemente el producto de los elementos de su diagonal principal) y $\tilde{w}^T \Sigma^{-1} \tilde{w} = v^T v$ (que es simplemente una suma de nm cuadrados), donde $v = P^{-1} \tilde{w}$. Las siguientes observaciones son interesantes:

- [1] La descomposición $\Sigma = PP^T$ sólo existe cuando Σ es definida positiva. Puede demostrarse [vid. Pagano (1973); Ansley (1979)] que si el proceso [2.1.3] es estacionario (en el sentido de [2.1.6]), entonces Σ es definida positiva, aunque el recíproco no es en general cierto (salvo para procesos $AR(p)$ puros). Luego siempre que el proceso [2.1.3] sea estacionario, es posible encontrar la descomposición $\Sigma = PP^T$, independientemente de si el proceso es o no invertible. Es importante señalar que, en este contexto, estamos hablando de estacionariedad *numérica*, es decir, de aquellos valores de los parámetros autorregresivos que hacen que la matriz Σ sea definida positiva.
- [2] La obtención de P requiere el cálculo de nm raíces cuadradas y de aproximadamente $(nm)^3/6$ multiplicaciones; además, la solución del sistema triangular $Pv = \tilde{w}$ requiere $(nm)^2/2$ multiplicaciones y nm divisiones adicionales. No obstante, cuando $p=0$, la estructura especial de la matriz Σ ($\Gamma(k)=0$ para $k > q$) permite obtener P , utilizando el algoritmo de Martin y Wilkinson (1965) con tan sólo nm raíces cuadradas y aproximadamente $nm(mq+1)(mq+2)/4$ multiplicaciones y resolver el sistema triangular $Pv = \tilde{w}$ con aproximadamente $(nm)(mq+1)$ multiplicaciones y nm divisiones adicionales. Esta ganancia es siempre interesante por cuanto que n debe ser bastante mayor que q .
- [3] Cuando el proceso no es invertible (en el sentido de [2.1.7]), el cálculo del determinante $|P|$ como el producto de los elementos de la diagonal principal de P , puede plantear problemas numéricos (*overflow*) cuando n es grande. No obstante, almacenar $|P|$ en la forma $a2^b$ [vid. Martin y Wilkinson (1965)] es suficiente para resolver el problema.

En definitiva, la evaluación directa de [2.2.1] permite calcular el valor de la *FVE* en cualquier punto del espacio paramétrico que satisfaga las condiciones de estacionariedad, aunque no necesariamente las de invertibilidad. No obstante, cuando $p \neq 0$, la matriz P contiene $nm(nm+1)/2$ elementos distintos que hay que almacenar simultáneamente; si n y/o m son grandes (n siempre debería serlo), la carga

computacional es muy elevada. Por ello, resulta conveniente disponer de otros métodos para evaluar $|\Sigma|$ y $\tilde{w}^T \Sigma^{-1} \tilde{w}$, que reduzcan tanto la carga computacional como los requisitos de almacenamiento.

2.2.2. La formulación de Nicholls y Hall (1979)

Siguiendo a Nicholls y Hall (1979), para obtener expresiones de $|\Sigma|$ y $\tilde{w}^T \Sigma^{-1} \tilde{w}$, empezamos escribiendo el proceso ARMA multivariante como en [2.1.17]:

$$D_{\Phi,n} \tilde{w} = D_{\Theta,n} a + Vu_* \quad [2.2.15]$$

Premultiplicando esta expresión por $D_{\Theta,n}^{-1}$ ($D_{\Theta,n}$ es no singular; *vid.* [2.1.20] y [2.1.23]), resulta que:

$$a = K\tilde{w} + Zu_* \quad [2.2.16]$$

donde:

$$K = D_{\Theta,n}^{-1} D_{\Phi,n} \quad [2.2.17]$$

$$Z = -D_{\Theta,n}^{-1} V \quad [2.2.18]$$

Nótese que K es $(nm \times nm)$ y Z es $(nm \times (p+q)m)$. Por tanto, podemos escribir el siguiente sistema:

$$\begin{bmatrix} u_* \\ a \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{((p+q)m \times nm)} \\ K \end{bmatrix} \tilde{w} + \begin{bmatrix} I_{(p+q) \times m} \\ Z \end{bmatrix} u_* \quad [2.2.19]$$

Definimos ahora dos matrices R ($m \times m$) y T ($(p+q)m \times (p+q)m$) tales que si:

$$E[a_i a_i^T] = \sigma^2 Q_{(m \times m)} \quad [2.2.20]$$

$$E[u_* u_*^T] = \sigma^2 \Omega_{((p+q)m \times (p+q)m)} \quad [2.2.21]$$

entonces:

$$RQR^T = I_m \quad (\Leftrightarrow Q^{-1} = R^T R) \quad [2.2.22]$$

$$T\Omega T^T = I_{(p+q)m} \quad (\Leftrightarrow \Omega^{-1} = T^T T) \quad [2.2.23]$$

Si premultiplicamos [2.2.19] por $\text{diag}[T, (\mathbf{I}_n \otimes \mathbf{R})]$ (matriz diagonal por bloques de orden $(p+q+n)m$), tenemos que:

$$\begin{bmatrix} \mathbf{T}u_* \\ (\mathbf{I}_n \otimes \mathbf{R})a \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ (\mathbf{I}_n \otimes \mathbf{R})K \end{bmatrix} \tilde{w} + \begin{bmatrix} \mathbf{I} \\ (\mathbf{I}_n \otimes \mathbf{R})ZT^{-1} \end{bmatrix} \mathbf{T}u_* \quad [2.2.24]$$

o bien:

$$\begin{bmatrix} e_* \\ e \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ (\mathbf{I}_n \otimes \mathbf{R})K \end{bmatrix} \tilde{w} + \begin{bmatrix} \mathbf{I} \\ (\mathbf{I}_n \otimes \mathbf{R})ZT^{-1} \end{bmatrix} e_* \quad [2.2.25]$$

o, finalmente:

$$\varepsilon = \mathbf{T}\tilde{w} + \Lambda e_* \quad [2.2.26]$$

donde ε ($(p+q+n)m \times 1$), e_* ($(p+q)m \times 1$), e ($nm \times 1$), \mathbf{T} ($(p+q+n)m \times nm$) y Λ ($(p+q+n)m \times (p+q)m$) se definen de acuerdo con [2.2.24]-[2.2.26]. Para obtener la función de densidad del vector ε , basta con tener en cuenta que:

$$E[e_* e_*^T] = E[\mathbf{T}u_* u_*^T \mathbf{T}^T] = \sigma^2 \mathbf{T} \mathbf{\Omega} \mathbf{T}^T = \sigma^2 \mathbf{I}_{(p+q)m} \quad [2.2.27]$$

$$E[e_* e^T] = E[\mathbf{T}u_* a^T (\mathbf{I}_n \otimes \mathbf{R})^T] = \mathbf{T} \mathbf{0} (\mathbf{I}_n \otimes \mathbf{R})^T = \mathbf{0}_{((p+q)m \times nm)} \quad [2.2.28]$$

$$E[ee^T] = E[(\mathbf{I}_n \otimes \mathbf{R})aa^T (\mathbf{I}_n \otimes \mathbf{R})^T] = \sigma^2 (\mathbf{I}_n \otimes \mathbf{R})(\mathbf{I}_n \otimes \mathbf{Q})(\mathbf{I}_n \otimes \mathbf{R})^T = \sigma^2 \mathbf{I}_{nm} \quad [2.2.29]$$

donde se han utilizado [2.2.22], [2.2.23] y [2.1.25], así como la independencencia entre a y u_* y [2.1.25]. Teniendo en cuenta [2.2.27]-[2.2.29], resulta evidente que:

$$E[\varepsilon \varepsilon^T] = E \begin{bmatrix} e_* e_*^T & e_* e^T \\ e e_*^T & e e^T \end{bmatrix} = \sigma^2 \mathbf{I}_{(p+q+n)m} \quad [2.2.30]$$

Finalmente, bajo la hipótesis $a_t \sim i.i.d. N(\mathbf{0}, \sigma^2 \mathbf{Q})$ para todo t , tenemos que la función de densidad del vector ε es la siguiente:

$$f(\varepsilon) = (2\pi\sigma^2)^{-\frac{1}{2}(p+q+n)m} \cdot \exp \left[-\frac{1}{2\sigma^2} \varepsilon^T \varepsilon \right] \quad [2.2.31]$$

donde, reescribiendo [2.2.26]:

$$\boldsymbol{\varepsilon} = (\boldsymbol{\Upsilon} \ \boldsymbol{\Lambda}) \begin{bmatrix} \tilde{\boldsymbol{w}} \\ \boldsymbol{e}_* \end{bmatrix} \quad [2.2.32]$$

Para expresar la función de densidad conjunta de \boldsymbol{w} y \boldsymbol{e}_* , tan sólo hay que calcular el jacobiano de la transformación [2.2.32], que será el valor absoluto del determinante $|(\boldsymbol{\Upsilon} \ \boldsymbol{\Lambda})|$. Teniendo en cuenta (vid. [2.2.25]-[2.2.26]) que:

$$(\boldsymbol{\Upsilon} \ \boldsymbol{\Lambda}) = \begin{bmatrix} \mathbf{0} & \boldsymbol{I} \\ (\boldsymbol{I}_n \otimes \boldsymbol{R})\boldsymbol{K} & (\boldsymbol{I}_n \otimes \boldsymbol{R})\boldsymbol{Z}\boldsymbol{T}^{-1} \end{bmatrix} \quad [2.2.33]$$

el determinante de esta matriz es:

$$-|(\boldsymbol{I}_n \otimes \boldsymbol{R})\boldsymbol{K}| = -|\boldsymbol{R}|^n \quad [2.2.34]$$

donde se ha tenido presente que, por [2.2.17], $|\boldsymbol{K}| = 1$. Además, utilizando [2.2.22], resulta evidente que $|\boldsymbol{R}\boldsymbol{Q}\boldsymbol{R}^T| = 1$, por lo que $|\boldsymbol{R}|^2 = |\boldsymbol{Q}|^{-1}$ y entonces $|\boldsymbol{R}| = |\boldsymbol{Q}|^{-1/2}$. Sustituyendo este resultado en [2.2.34], el jacobiano de la transformación [2.2.32] resulta ser $|\boldsymbol{Q}|^{-n/2}$. Por tanto, la función de densidad conjunta de \boldsymbol{w} y \boldsymbol{e}_* es la siguiente:

$$f(\boldsymbol{w}, \boldsymbol{e}_*) = (2\pi\sigma^2)^{-\frac{1}{2}(p+q+n)m} \cdot |\boldsymbol{Q}|^{-\frac{n}{2}} \cdot \exp \left[-\frac{1}{2\sigma^2} (\boldsymbol{\Upsilon}\tilde{\boldsymbol{w}} + \boldsymbol{\Lambda}\boldsymbol{e}_*)^T (\boldsymbol{\Upsilon}\tilde{\boldsymbol{w}} + \boldsymbol{\Lambda}\boldsymbol{e}_*) \right] \quad [2.2.35]$$

Para obtener la función de densidad de \boldsymbol{w} , empezamos calculando el estimador por máxima verosimilitud del vector \boldsymbol{e}_* en el modelo [2.2.26] (reescrito $\boldsymbol{\Upsilon}\tilde{\boldsymbol{w}} = -\boldsymbol{\Lambda}\boldsymbol{e}_* + \boldsymbol{\varepsilon}$). Evidentemente, dicho estimador está dado por:

$$\hat{\boldsymbol{e}}_* = -(\boldsymbol{\Lambda}^T \boldsymbol{\Lambda})^{-1} \boldsymbol{\Lambda}^T \boldsymbol{\Upsilon} \tilde{\boldsymbol{w}} \quad [2.2.36]$$

A partir de esta expresión, resulta inmediato comprobar que:

$$(\boldsymbol{\Upsilon}\tilde{\boldsymbol{w}} + \boldsymbol{\Lambda}\boldsymbol{e}_*)^T (\boldsymbol{\Upsilon}\tilde{\boldsymbol{w}} + \boldsymbol{\Lambda}\boldsymbol{e}_*) = (\boldsymbol{\Upsilon}\tilde{\boldsymbol{w}} + \boldsymbol{\Lambda}\hat{\boldsymbol{e}}_*)^T (\boldsymbol{\Upsilon}\tilde{\boldsymbol{w}} + \boldsymbol{\Lambda}\hat{\boldsymbol{e}}_*) + (\boldsymbol{e}_* - \hat{\boldsymbol{e}}_*)^T (\boldsymbol{\Lambda}^T \boldsymbol{\Lambda}) (\boldsymbol{e}_* - \hat{\boldsymbol{e}}_*) \quad [2.2.37]$$

ya que:

$$(\mathbf{T}\tilde{\mathbf{w}} + \mathbf{\Lambda}\hat{\mathbf{e}}_*)^T(\mathbf{T}\tilde{\mathbf{w}} + \mathbf{\Lambda}\hat{\mathbf{e}}_*) = \tilde{\mathbf{w}}^T[\mathbf{T}^T\mathbf{T} - \mathbf{T}^T\mathbf{\Lambda}(\mathbf{\Lambda}^T\mathbf{\Lambda})^{-1}\mathbf{\Lambda}^T\mathbf{T}]\tilde{\mathbf{w}} \quad [2.2.38]$$

$$(\mathbf{e}_* - \hat{\mathbf{e}}_*)^T(\mathbf{\Lambda}^T\mathbf{\Lambda})(\mathbf{e}_* - \hat{\mathbf{e}}_*) = \mathbf{e}_*^T(\mathbf{\Lambda}^T\mathbf{\Lambda})\mathbf{e}_* + 2\tilde{\mathbf{w}}^T\mathbf{T}^T\mathbf{\Lambda}\mathbf{e}_* + \tilde{\mathbf{w}}^T\mathbf{T}^T\mathbf{\Lambda}(\mathbf{\Lambda}^T\mathbf{\Lambda})^{-1}\mathbf{\Lambda}^T\mathbf{T}\tilde{\mathbf{w}} \quad [2.2.39]$$

Por tanto, podemos factorizar [2.2.35] como sigue:

$$\begin{aligned} f(\mathbf{w}, \mathbf{e}_*) &= (2\pi\sigma^2)^{-\frac{nm}{2}} \cdot |\mathbf{Q}|^{-\frac{n}{2}} \cdot |\mathbf{\Lambda}^T\mathbf{\Lambda}|^{-\frac{1}{2}} \cdot \exp\left[-\frac{1}{2\sigma^2}(\mathbf{T}\tilde{\mathbf{w}} + \mathbf{\Lambda}\hat{\mathbf{e}}_*)^T(\mathbf{T}\tilde{\mathbf{w}} + \mathbf{\Lambda}\hat{\mathbf{e}}_*)\right] \times \\ &\quad \times (2\pi\sigma^2)^{-\frac{1}{2}(p+q)m} \cdot |\mathbf{\Lambda}^T\mathbf{\Lambda}|^{\frac{1}{2}} \cdot \exp\left[-\frac{1}{2\sigma^2}(\mathbf{e}_* - \hat{\mathbf{e}}_*)^T(\mathbf{\Lambda}^T\mathbf{\Lambda})(\mathbf{e}_* - \hat{\mathbf{e}}_*)\right] \\ &= f(\mathbf{w}) \times f(\mathbf{e}_*|\mathbf{w}) \end{aligned} \quad [2.2.40]$$

Entonces:

$$f(\mathbf{w}) = (2\pi\sigma^2)^{-\frac{nm}{2}} \cdot |\mathbf{Q}|^{-\frac{n}{2}} \cdot |\mathbf{\Lambda}^T\mathbf{\Lambda}|^{-\frac{1}{2}} \cdot \exp\left[-\frac{1}{2\sigma^2}(\mathbf{T}\tilde{\mathbf{w}} + \mathbf{\Lambda}\hat{\mathbf{e}}_*)^T(\mathbf{T}\tilde{\mathbf{w}} + \mathbf{\Lambda}\hat{\mathbf{e}}_*)\right] \quad [2.2.41]$$

Considerando [2.2.41] como función de los parámetros Φ_i ($i=1,2,\dots,p$), Θ_i ($i=1,2,\dots,q$), μ , σ^2 y \mathbf{Q} que en ella intervienen, obtenemos finalmente la FVE para una muestra \mathbf{w} dada, que es:

$$L(\Phi_i, \Theta_i, \mu, \sigma^2, \mathbf{Q}|\mathbf{w}) = (2\pi\sigma^2)^{-\frac{nm}{2}} \cdot |\mathbf{Q}|^{-\frac{n}{2}} \cdot |\mathbf{\Lambda}^T\mathbf{\Lambda}|^{-\frac{1}{2}} \cdot \exp\left[-\frac{1}{2\sigma^2}(\mathbf{T}\tilde{\mathbf{w}} + \mathbf{\Lambda}\hat{\mathbf{e}}_*)^T(\mathbf{T}\tilde{\mathbf{w}} + \mathbf{\Lambda}\hat{\mathbf{e}}_*)\right]$$

[2.2.42]

La función de verosimilitud sólo depende, para una muestra dada, de los parámetros mencionados. Para verlo, basta tener en cuenta la definición de las matrices \mathbf{T} y $\mathbf{\Lambda}$ en [2.2.33] y la definición del vector $\hat{\mathbf{e}}_*$ en [2.2.36]. Tan sólo resta obtener una expresión para la matriz $\mathbf{\Omega}$ de [2.2.21] y [2.2.23] en función de los parámetros del modelo ARMA multivariante. Dicha matriz puede partitionarse de la siguiente manera:

$$\mathbf{\Omega} = \frac{1}{\sigma^2} E[\mathbf{u}_* \mathbf{u}_*^T] = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix} \quad [2.2.43]$$

donde:

$$\sigma^2 \mathbf{A}_{(pm \times pm)} = E[\tilde{\mathbf{w}}_* \tilde{\mathbf{w}}_*^T] \quad [2.2.44]$$

$$\sigma^2 \mathbf{B}_{(pm \times qm)} = E[\tilde{\mathbf{w}}_* \mathbf{a}_*^T] \quad [2.2.45]$$

$$\sigma^2 \mathbf{C}_{(qm \times qm)} = E[\mathbf{a}_* \mathbf{a}_*^T] \quad [2.2.46]$$

Como $A_{ij} = (1/\sigma^2)E[\tilde{\mathbf{w}}_{i-p} \tilde{\mathbf{w}}_{j-p}^T] = \Gamma(j-i)$ y $A_{ji} = (1/\sigma^2)E[\tilde{\mathbf{w}}_{j-p} \tilde{\mathbf{w}}_{i-p}^T] = \Gamma(i-j) = \Gamma(j-i)^T = A_{ij}^T$ ($i, j=1, 2, \dots, p$), para obtener la matriz \mathbf{A} necesitamos calcular $\Gamma(k)$ desde $k=0$ hasta $k=p-1$ (p matrices distintas). Además, $B_{ij} = (1/\sigma^2)E[\tilde{\mathbf{w}}_{i-p} \mathbf{a}_{j-q}^T] = \Gamma_{wa}(j-i-q+p)$ ($i=1, 2, \dots, p; j=1, 2, \dots, q$); por tanto, para obtener \mathbf{B} necesitamos calcular $\Gamma_{wa}(k)$ desde $k=-q+1$ hasta $k=p-1$ ($p+q-1$ matrices distintas). Por último, es evidente que:

$$\mathbf{C} = (\mathbf{I}_q \otimes \mathbf{Q}) \quad [2.2.47]$$

Todos los cálculos referentes a las matrices \mathbf{A} y \mathbf{B} pueden llevarse a cabo según [2.2.8]-[2.2.14] y [2.2.4]-[2.2.6], respectivamente. Llegados a este punto, la evaluación de la FVE [2.2.42] en un punto del espacio paramétrico (definido por unos valores numéricos concretos de Φ_i ($i=1, 2, \dots, p$), Θ_i ($i=1, 2, \dots, q$), μ , σ^2 y \mathbf{Q}) puede llevarse a cabo mediante el siguiente procedimiento (algoritmo [NH]):

[NH.1] Calcular $\Gamma_{wa}(k)$ según [2.2.4]-[2.2.6] ($k=0, 1, \dots, q$).

[NH.2] Calcular $W(k)$ según [2.2.10] ($k=0, 1, \dots, p$).

[NH.3] Formar el sistema [2.2.12] y resolverlo para $\gamma(k)$ ($k=0, 1, \dots, p$).

[NH.4] Recuperar las matrices $\Gamma(k)$ ($k=0, 1, \dots, p$) a partir de los vectores $\gamma(k)$ ($k=0, 1, \dots, p$).

Estas cuatro operaciones generan la matriz Ω de [2.2.43], según lo expuesto anteriormente. Para evaluar los determinantes y la forma cuadrática de [2.2.42], resulta interesante escribir esta última (*vid.* [2.2.38]) como:

$$(\mathbf{T}\tilde{\mathbf{w}} + \Lambda \hat{\mathbf{e}}_*)^T (\mathbf{T}\tilde{\mathbf{w}} + \Lambda \hat{\mathbf{e}}_*) = (\mathbf{T}\tilde{\mathbf{w}})^T [\mathbf{I}_{nm} - \Lambda (\Lambda^T \Lambda)^{-1} \Lambda^T] (\mathbf{T}\tilde{\mathbf{w}}) \quad [2.2.48]$$

[NH.5] Obtener \mathbf{R} (y, simultáneamente, $|\mathbf{Q}|$) según [2.2.22] y \mathbf{T}^{-1} según [2.2.23].

[NH.6] Calcular el vector $\mathbf{T}\tilde{\mathbf{w}}$ y la matriz Λ según [2.2.25]-[2.2.26].

[NH.7] Evaluar $|\Lambda^T \Lambda|$ y la forma cuadrática [2.2.48].

Respecto a [NH.5], es suficiente tener presente que $\mathbf{Q} = \mathbf{Q}_1 \mathbf{Q}_1^T$ (su descomposición de Cholesky), por lo que, según [2.2.22], $\mathbf{R} = \mathbf{Q}_1^{-1}$ (y $|\mathbf{Q}| = |\mathbf{Q}_1|^2$, que es simplemente el producto de los elementos de la diagonal principal de \mathbf{Q}_1 al cuadrado); análogamente, $\Omega = \Omega_1 \Omega_1^T$, por lo que, según [2.2.23],

$T^{-1} = \Omega_1$ (nótese que tan sólo se necesita T^{-1} , no T). Ya que Q es *siempre* definida positiva (vid. [2.1.11]), su descomposición de Cholesky siempre existe; respecto a la descomposición de Cholesky de Ω , existirá siempre que Ω sea definida positiva, para lo cual es condición suficiente que el proceso ARMA cuyos parámetros se desea estimar sea estacionario [vid. Pagano (1973); Dent (1977); Ansley (1979)]. Por tanto, el paso [NH.5] proporciona una comprobación necesaria para la estacionariedad del proceso.

Respecto a [NH.6] y a la vista de [2.2.25], sólo es necesario calcular $(I_n \otimes R)K\tilde{w}$ y $(I_n \otimes R)ZT^{-1}$. Hall y Nicholls (1980) describen cómo calcular recursivamente $(I_n \otimes R)K\tilde{w}$ y $(I_n \otimes R)Z$; también proponen evaluar $(\mathbf{T}\tilde{w} + \Lambda\hat{e}_*)^T(\mathbf{T}\tilde{w} + \Lambda\hat{e}_*)$ directamente (haciendo uso de [2.2.36]) o utilizando [2.2.48], que es la suma de residuos al cuadrado de la regresión de $\mathbf{T}\tilde{w}$ sobre Λ . Sin embargo, este método tiene el inconveniente de requerir una gran cantidad de almacenamiento (en concreto, los $(p+q+n)m \times (p+q)m$ elementos de Λ). Esta cuestión, así como la velocidad de cálculo, pueden mejorarse considerablemente explorando con mayor profundidad la estructura de la *FVE*, tal y como se hace en la Sección 2.3.

2.2.3. Técnicas basadas en las representaciones en *espacio de los estados*

Las contribuciones más recientes sobre la evaluación de la *FVE* de procesos ARMA estacionarios, están invariablemente basadas en alguna representación en *espacio de los estados* (*EE* en adelante) del proceso cuyos parámetros se desea estimar. En concreto, la utilización de los filtros de Kalman y de Chandrasekhar, como un medio computacionalmente eficiente para evaluar la *FVE*, puede encontrarse en Gardner, Harvey y Phillips (1980), Pearlman (1980), Melard (1984) y Kohn y Ansley (1985) en el contexto univariante y en Ansley y Kohn (1983) y Shea (1984, 1987, 1989) para procesos multivariantes.

En Shea (1989) se propone un algoritmo para evaluar la *FVE*, que puede considerarse como el más eficiente (y, hasta donde conocemos, el más reciente) disponible en la actualidad. Además, para calibrar el interés práctico del nuevo algoritmo de la sección 2.3 respecto a los ya existentes, en la sección 2.4 se presentan algunas pruebas comparativas de las prestaciones de distintos algoritmos [incluido, lógicamente, el de Shea (1989)]. Por todo ello, puede resultar útil resumir este trabajo. El proceso [2.1.3] puede escribirse en *EE* como:

$$\beta_t = T\beta_{t-1} + Ra_{t-1} \quad [2.2.49]$$

$$\tilde{w}_t = H^T\beta_t + a_t \quad [2.2.50]$$

donde β_t es el vector de estado de dimensión gm ($g = \text{máximo}\{p, q\}$), [2.2.49] es la ecuación de estado, [2.2.50] es la ecuación de observación [vid. Anderson y Moore (1979)] y:

$$T = \begin{pmatrix} \Phi_1 & I & 0 & \cdots & 0 \\ \Phi_2 & 0 & I & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Phi_{g-1} & 0 & 0 & \cdots & I \\ \Phi_g & 0 & 0 & \cdots & 0 \end{pmatrix} \quad R = \begin{pmatrix} \Phi_1 - \Theta_1 \\ \Phi_2 - \Theta_2 \\ \vdots \\ \Phi_g - \Theta_g \end{pmatrix} \quad H = \begin{pmatrix} I_m \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad [2.2.51]$$

donde $\Phi_i = 0$ para $i > p$ y $\Theta_i = 0$ para $i > q$ (las matrices R y T de [2.2.49] y [2.2.51] no deben confundirse con las matrices R y T de [2.2.22] y [2.2.23], respectivamente). Si definimos $b_{t|t-1}$ como el estimador lineal con error cuadrático medio mínimo de β_t , dada toda la información hasta $t-1$, y el proceso de innovaciones como $v_t = \tilde{w}_t - E[\tilde{w}_t | \tilde{w}_{t-1}, \tilde{w}_{t-2}, \dots, \tilde{w}_1] = \tilde{w}_t - H^T b_{t|t-1}$, puede demostrarse que $E[v_t] = 0$, $E[v_t v_t^T] = \sigma^2 F_t$ y que $E[v_t v_{t+k}^T] = 0$ para $k \neq 0$. Además, la FVE de la muestra considerada está dada por:

$$L(\Phi_i, \Theta_i, \mu, \sigma^2, Q | w) = (2\pi\sigma^2)^{-\frac{nm}{2}} \cdot \left[\prod_{t=1}^n |F_t| \right]^{-\frac{1}{2}} \cdot \exp \left[-\frac{1}{2\sigma^2} \sum_{t=1}^n v_t^T F_t^{-1} v_t \right]$$

[2.2.52]

Por tanto, la clave para evaluar eficientemente la FVE consiste en disponer de un procedimiento eficiente para evaluar las innovaciones v_t junto con sus matrices de covarianzas F_t . Dicho procedimiento está descrito por el siguiente conjunto de ecuaciones recursivas (ecuaciones de Chandrasekhar):

$$\begin{aligned} v_t &= \tilde{w}_t - H^T b_{t|t-1} \\ b_{t|t-1} &= T b_{t-1|t-2} + K_{t-1} F_{t-1}^{-1} v_{t-1} \\ K_t &= K_{t-1} - T L_{t-1} M_{t-1} L_{t-1}^T H \\ F_t &= F_{t-1} - H^T L_{t-1} M_{t-1} L_{t-1}^T H \\ M_t &= M_{t-1} + M_{t-1} L_{t-1}^T H F_t^{-1} H^T L_{t-1} M_{t-1} \\ L_t &= T L_{t-1} - K_{t-1} F_{t-1}^{-1} H^T L_{t-1} \end{aligned} \quad [2.2.53]$$

donde M_t es $(m \times m)$, L_t es $(gm \times m)$ y $K_t = E[\beta_{t+1} v_t^T]$ es la ganancia del filtro de Kalman. Las condiciones iniciales para las ecuaciones [2.2.53] están dadas por:

$$\mathbf{b}_{1|0} = \mathbf{0} ; \mathbf{K}_1 = \mathbf{L}_1 = \mathbf{T}\mathbf{P}_{1|0}\mathbf{H} + \mathbf{R}\mathbf{Q} ; \mathbf{F}_1 = \mathbf{H}^T\mathbf{P}_{1|0}\mathbf{H} + \mathbf{Q} ; \mathbf{M}_1 = -\mathbf{F}_1^{-1} \quad [2.2.54]$$

donde, considerando $\mathbf{P}_{1|0}\mathbf{H} = E[\beta_t\beta_t^T]\mathbf{H}$ como un vector de g matrices ($m \times m$), puede demostrarse que la i -ésima componente de $\mathbf{P}_{1|0}\mathbf{H}$ está dada por:

$$(\mathbf{P}_{1|0}\mathbf{H})_i = \sum_{j=i}^p \Phi_j \Gamma(j-i+1) - \sum_{j=i}^q \Theta_j \Gamma_{wa}(i-j+1)^T, \quad i=1,2,\dots,g \quad [2.2.55]$$

(vid. apartado 2.2.1 para el cálculo de las matrices de covarianzas que aparecen en [2.2.55]).

En este contexto, el proceso ARMA es estacionario cuando los autovalores de \mathbf{T} están dentro del círculo unitario (condición equivalente a [2.1.6]). Análogamente, cuando los autovalores de $\mathbf{T} - \mathbf{R}\mathbf{H}^T$ satisfacen esta misma condición, el proceso es invertible (condición equivalente a [2.1.7]). Para un proceso estacionario e invertible, puede demostrarse que el filtro descrito por las ecuaciones [2.2.53] alcanza un *estado estacionario* con $\mathbf{K}_t \rightarrow \mathbf{R}\mathbf{Q}$, $\mathbf{F}_t \rightarrow \mathbf{Q}$ y $\mathbf{v}_t \rightarrow \mathbf{a}_t$ en media cuadrática. En la práctica, esto ocurre a medida que t en [2.2.53] se va haciendo más grande. Una vez alcanzada esta situación, las restantes matrices \mathbf{F}_t pueden igualarse a \mathbf{Q} y las innovaciones \mathbf{v}_t pueden calcularse recursivamente a partir de la forma [2.2.2] del proceso ARMA como:

$$\mathbf{v}_t = \tilde{\mathbf{w}}_t - \sum_{j=1}^p \Phi_j \tilde{\mathbf{w}}_{t-j} + \sum_{j=1}^q \Theta_j \mathbf{v}_{t-j} \quad [2.2.56]$$

Esta forma de proceder se conoce como "*switching to quick recursions*" (para un proceso AR(p), las ecuaciones [2.2.53] convergen *exactamente* en p recursiones).

En la práctica, puede considerarse que el filtro descrito por [2.2.53] ha alcanzado un estado estacionario cuando:

$$\text{máximo} \left[\frac{|\mathbf{F}_t(i,i) - \mathbf{Q}(i,i)|}{\mathbf{Q}(i,i)} \right] < \delta, \quad i=1,2,\dots,m \quad [2.2.57]$$

El criterio de convergencia (representado por δ), puede hacerse lo suficientemente rígido como para que la diferencia $\mathbf{F}_t - \mathbf{Q}$ sea despreciable, de forma que la diferencia entre el valor de la FVE y el resultado de esta estrategia sea también despreciable (el valor de la función de verosimilitud es cada vez más *exacto* a medida que $\delta \rightarrow 0$). Si, cuando ocurre, no se reconoce esta convergencia, continuar con las

recursiones [2.2.53] hasta $t=n$ implica una gran pérdida de tiempo y puede provocar problemas numéricos (*underflow*) cuando n es grande.

Shea (1989) examina tres criterios de convergencia: (i) $\delta < 0$ (*FVE*), (ii) $\delta = 10^{-3}$ (función de verosimilitud *aproximada*) y (iii) $\delta = \infty$ (función de verosimilitud *condicional*). Los resultados de este examen ilustran los obtenidos por Dent y Min (1978) y Ansley y Newbold (1980) en el contexto univariante (*vid.* Capítulo 3 para más detalles).

Finalmente, es importante señalar dos aspectos. Primero, aunque el procedimiento descrito es más eficiente que el propuesto por Kohn y Ansley (1983), basado en el filtro de Kalman [excepto para procesos AR(1), AR(2) y ARMA(2,1); *vid.* Shea (1987)], no puede ser empleado cuando faltan observaciones en la muestra considerada (en el caso de observaciones ausentes o de datos agregados temporalmente), ya que, para afrontar esta situación, es necesario permitir que las matrices \mathbf{R} , \mathbf{T} y \mathbf{H} de [2.2.49]-[2.2.51] varíen en el tiempo y las recursiones [2.2.53] están derivadas bajo el supuesto de que \mathbf{R} , \mathbf{T} y \mathbf{H} son constantes a lo largo del período muestral. Por último, estrictamente hablando, este procedimiento *no* permite calcular directamente (como un subproducto) los residuos correspondientes a la muestra y a los parámetros estimados, cuya utilidad reside en la diagnosis del modelo estimado y en el cálculo de previsiones de valores futuros.

2.2.4. La aproximación de Hillmer y Tiao (1979)

Hillmer y Tiao (1979) han demostrado que una aproximación adecuada a la *FVE* de procesos ARMA(p,q) puede obtenerse mediante el siguiente procedimiento (algoritmo [HT]):

[HT.1] Evaluar la secuencia $\mathbf{u}_t = \tilde{\mathbf{w}}_t - \sum_{j=1}^p \Phi_j \tilde{\mathbf{w}}_{t-j}$ ($t = p+1, p+2, \dots, n$) considerando $\tilde{\mathbf{w}}_t$ ($t = 1, 2, \dots, p$) como números fijos.

[HT.2] Considerar la secuencia \mathbf{u}_t como $n-p$ observaciones generadas por un proceso MA(q).

[HT.3] Utilizar dicha secuencia en el cálculo de la *FVE* de un proceso MA(q) puro.

La forma de evaluar la *FVE* de un proceso MA(q) utilizada por Hillmer y Tiao (1979) (paso [HT.3]) es matemáticamente equivalente a las propuestas de Ljung y Box (1979) y de Nicholls y Hall (1979) (*vid.* nota [1] al final de la Sección 2.3). Por otro lado, aunque el algoritmo [HT] resulta adecuado (*vid.* Capítulo 3) en muchas ocasiones e implica una evaluación computacionalmente más

eficiente que una evaluación analíticamente exacta (cuando el proceso contiene parte AR), es importante subrayar tres inconvenientes que pueden surgir en su aplicación práctica: (i) la ausencia de detección, como subproducto de ciertos cálculos, de situaciones de no estacionariedad, (ii) la imposibilidad de estimar los primeros p residuos y (iii) a consecuencia de ello, la imposibilidad de detectar anomalías (posiblemente importantes) en la muestra cuando tienen lugar dentro de las primeras p observaciones.

Como se ilustra en el Capítulo 3, no es difícil encontrar situaciones prácticas en las que alguna de estas tres circunstancias produce estimaciones de los parámetros que pueden conducir a conclusiones erróneas. No obstante, en el caso de procesos MA(q) puros o cuando no tienen lugar los inconvenientes citados anteriormente, el procedimiento de Hillmer y Tiao (1979) puede resultar adecuado.

2.3. Un nuevo procedimiento para evaluar la función de verosimilitud exacta

El algoritmo que se presenta en esta sección (algoritmo [FVE]) tiene su origen en la formulación de la FVE de Nicholls y Hall (1979) (*vid.* apartado 2.2.2). El resultado permite demostrar la equivalencia matemática entre la formulación de Nicholls y Hall (1979) y la extensión multivariante de la formulación de Ljung y Box (1979). Esta equivalencia permite, además, interpretar de distintas formas los factores que aparecen en la FVE [2.2.42]. También es posible demostrar la equivalencia matemática entre las formulaciones de Nicholls y Hall (1979) y de Hillmer y Tiao (1979)¹. Por tanto, el algoritmo [FVE] engloba las formulaciones de la FVE más utilizadas *entre aquellas que no se basan en alguna representación en EE*. Además (*vid.* sección 2.4) el algoritmo [FVE] puede competir (y en muchos casos resultar superior) en términos de eficiencia computacional con los algoritmos más eficientes basados en alguna representación en EE [en concreto, Shea (1989)].

2.3.1. El algoritmo [FVE]

Una primera cuestión relevante para evaluar eficientemente la FVE [2.2.42] consiste en escribir la forma cuadrática [2.2.48] como:

$$(\mathbf{Y}\tilde{\mathbf{w}} + \mathbf{\Lambda}\hat{\mathbf{e}}_*)^T(\mathbf{Y}\tilde{\mathbf{w}} + \mathbf{\Lambda}\hat{\mathbf{e}}_*) = (\mathbf{Y}\tilde{\mathbf{w}})^T(\mathbf{Y}\tilde{\mathbf{w}}) - (\mathbf{\Lambda}^T\mathbf{Y}\tilde{\mathbf{w}})^T(\mathbf{\Lambda}^T\mathbf{\Lambda})^{-1}(\mathbf{\Lambda}^T\mathbf{Y}\tilde{\mathbf{w}}) \quad [2.3.1]$$

De esta forma, sólo es necesario calcular y almacenar los vectores $\mathbf{T}\tilde{\mathbf{w}}$ $((p+q+n)m \times 1)$, $\mathbf{\Lambda}^T \mathbf{T}\tilde{\mathbf{w}}$ $((p+q)m \times 1)$ y la matriz $\mathbf{\Lambda}^T \mathbf{\Lambda}$ $((p+q)m \times (p+q)m)$. Como los $(p+q)m$ primeros elementos de $\mathbf{T}\tilde{\mathbf{w}}$ son cero (vid. [2.2.25]) y la matriz $\mathbf{\Lambda}^T \mathbf{\Lambda}$ es simétrica, tan sólo hay que calcular y almacenar $[nm] + [(p+q)m] + [(p+q)m((p+q)m+1)/2]$ elementos distintos.

Para evaluar los nm elementos no nulos del vector $\mathbf{T}\tilde{\mathbf{w}}$, necesitamos calcular primero el vector $\mathbf{K}\tilde{\mathbf{w}}$. Si definimos [en la notación de Ljung y Box (1979)] $\hat{\mathbf{a}}_0 = E[\mathbf{a} | \mathbf{w}, \mathbf{u}_* = \mathbf{0}]$, es evidente, por [2.2.16], que $\mathbf{K}\tilde{\mathbf{w}} = \hat{\mathbf{a}}_0$. Por tanto, teniendo en cuenta la expresión [2.2.2] del proceso ARMA multivariante, los n vectores $(m \times 1)$ que componen $\hat{\mathbf{a}}_0$ pueden calcularse recursivamente como:

$$\hat{\mathbf{a}}_{0i} = \tilde{\mathbf{w}}_i - \sum_{j=1}^p \Phi_j \tilde{\mathbf{w}}_{i-j} + \sum_{j=1}^q \Theta_j \hat{\mathbf{a}}_{0,i-j}, \quad i=1,2,\dots,n \quad [2.3.2]$$

con $\tilde{\mathbf{w}}_i = \mathbf{0}$ para $i < 1$ y $\hat{\mathbf{a}}_{0i} = \mathbf{0}$ para $i < 1$. Entonces, cada uno de los n vectores η_i $(m \times 1)$ que componen el vector $\boldsymbol{\eta} = (\mathbf{I}_n \otimes \mathbf{R}) \mathbf{K}\tilde{\mathbf{w}}$ están dados por $\eta_i = \mathbf{R}\hat{\mathbf{a}}_{0i}$ ($i=1,2,\dots,n$).

Para evaluar los $(p+q)m((p+q)m+1)/2$ elementos distintos de $\mathbf{\Lambda}^T \mathbf{\Lambda}$, hay que tener en cuenta que, por [2.2.33]:

$$\mathbf{\Lambda}^T \mathbf{\Lambda} = \mathbf{I}_{(p+q)m} + \mathbf{T}^{-1T} \mathbf{X} \mathbf{T}^{-1} \quad [2.3.3]$$

donde:

$$\mathbf{X} = \mathbf{Z}^T (\mathbf{I}_n \otimes \mathbf{R}^T) (\mathbf{I}_n \otimes \mathbf{R}) \mathbf{Z} \quad [2.3.4]$$

o, por [2.2.18]:

$$\mathbf{X} = \mathbf{V}^T \mathbf{D}_{\boldsymbol{\theta},n}^{-1T} (\mathbf{I}_n \otimes \mathbf{R}^T) (\mathbf{I}_n \otimes \mathbf{R}) \mathbf{D}_{\boldsymbol{\theta},n}^{-1} \mathbf{V} \quad [2.3.5]$$

o, finalmente:

$$\mathbf{X} = \mathbf{V}_1^T \mathbf{H}^T \mathbf{H} \mathbf{V}_1 \quad [2.3.6]$$

donde, si definimos $g = \text{máximo}\{p, q\}$, \mathbf{V}_1 $(gm \times (p+q)m)$ contiene las primeras gm filas de \mathbf{V} y \mathbf{H} $(nm \times gm)$ contiene las primeras gm columnas de $(\mathbf{I}_n \otimes \mathbf{R}) \mathbf{D}_{\boldsymbol{\theta},n}^{-1}$. Teniendo en cuenta [2.1.23], resulta inmediato comprobar que $\mathbf{H} = (\mathbf{H}_1 \mathbf{H}_2 \cdots \mathbf{H}_g)$, con cada \mathbf{H}_i $(nm \times m)$ de la forma:

$$H_1 = \begin{bmatrix} R\Xi_0 \\ R\Xi_1 \\ \vdots \\ R\Xi_{n-1} \end{bmatrix}; H_2 = \begin{bmatrix} \mathbf{0} \\ R\Xi_0 \\ \vdots \\ R\Xi_{n-2} \end{bmatrix}; \dots; H_g = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ R\Xi_{n-g} \end{bmatrix} \quad [2.3.7]$$

donde, por [2.1.24], las n matrices Ξ_k ($k = 0, 1, \dots, n-1$) se calculan recursivamente como sigue:

$$\Xi_k = \sum_{j=1}^q \Theta_j \Xi_{k-j}, \quad k=1, 2, \dots, n-1 \quad [2.3.8]$$

con $\Xi_0 = I_m$ y $\Xi_k = \mathbf{0}$ para $k < 0$. Si definimos $P = H^T H$ ($gm \times gm$), para evaluar [2.3.6] sólo necesitamos calcular las $g(g+1)/2$ matrices P_{ij} ($i=1, 2, \dots, g; j=1, 2, \dots, i$) que componen la diagonal principal y el triángulo inferior de $H^T H$. Pero esta operación tan sólo requiere el cálculo de P_{i1} ($i=1, 2, \dots, g$) como:

$$P_{i1} = \sum_{k=0}^{n-i} \Xi_k^T R^T R \Xi_{k+i-1}, \quad i=1, 2, \dots, g \quad [2.3.9]$$

ya que, como puede comprobarse fácilmente, las demás matrices P_{ij} pueden calcularse recursivamente como:

$$P_{ij} = P_{i-1, j-1} - \Xi_{n-i+1}^T R^T R \Xi_{n-j+1}, \quad i=2, 3, \dots, g, \quad j=2, 3, \dots, i \quad [2.3.10]$$

Por tanto, una vez evaluada la matriz $H^T H$, podemos evaluar [2.3.6] y [2.3.3] para obtener los elementos requeridos de la matriz $\Lambda^T \Lambda$. Esta tarea puede llevarse a cabo teniendo en cuenta que [2.3.3] también puede escribirse como:

$$\Lambda^T \Lambda = I_{(p+q)m} + A^T H^T H A \quad [2.3.11]$$

donde los elementos de la matriz ($gm \times (p+q)m$) $A = V_1 T^{-1}$ se calculan de la siguiente manera:

$$A_{ij} = \sum_{k=0}^{p-i} \Phi_{p-k} T_{k+i, j}^{-1} - \sum_{k=0}^{q-i} \Theta_{q-k} T_{k+p+i, j}^{-1}, \quad i=1, 2, \dots, g, \quad j=1, 2, \dots, p+q \quad [2.3.12]$$

Por último, necesitamos calcular los $(p+q)m$ elementos del vector $\Lambda^T \mathbf{Y} \tilde{\mathbf{w}}$. Teniendo en cuenta [2.2.33]:

$$\Lambda^T \Upsilon \tilde{w} = T^{-1T} Z^T (I_n \otimes R^T) \eta \quad [2.3.13]$$

donde el vector $(nm \times 1)$ $\eta = (I_n \otimes R) K \tilde{w}$ ya se evaluó para calcular $\Upsilon \tilde{w}$. Además, por [2.2.18]:

$$\Lambda^T \Upsilon \tilde{w} = -T^{-1T} V^T D_{\Theta, n}^{-1T} (I_n \otimes R^T) \eta \quad [2.3.14]$$

o bien:

$$\Lambda^T \Upsilon \tilde{w} = -T^{-1T} g \quad [2.3.15]$$

donde:

$$g = V^T h \quad [2.3.16]$$

$$h = D_{\Theta, n}^{-1T} (I_n \otimes R^T) \eta \quad [2.3.17]$$

Los $p+q$ vectores $(m \times 1)$ g_j que componen el vector g pueden calcularse como:

$$g_j = \begin{cases} \sum_{i=1}^j \Phi_{p-j+i}^T h_i & , j=1, 2, \dots, p \\ -\sum_{i=1}^{j-p} \Theta_{q-j+i+p}^T h_i & , j=p+1, p+2, \dots, p+q \end{cases} \quad [2.3.18]$$

donde los g vectores $(m \times 1)$ h_j , necesarios en esta expresión, se calculan como:

$$h_j = \sum_{i=0}^{n-j} \Xi_i^T R^T \eta_{i+j} , j=1, 2, \dots, g \quad [2.3.19]$$

Nótese que, aunque [2.3.16] sugiere calcular h_j desde $j=1$ hasta $j=n$, la expresión [2.3.18] deja bien claro que dicho cálculo sólo es necesario desde $j=1$ hasta $j=g$. Una vez calculado el vector g , los elementos de $\Lambda^T \Upsilon \tilde{w}$ pueden evaluarse directamente de [2.3.15]. No obstante, por [2.3.15] y [2.3.16]:

$$\Lambda^T \Upsilon \tilde{w} = -T^{-1T} V^T h = -T^{-1T} V_1^T \tilde{h} = A^T \tilde{h} \quad [2.3.20]$$

donde A es la matriz de [2.3.11]-[2.3.12] y $\tilde{h} = (h_1^T, h_2^T, \dots, h_g^T)^T$. Por tanto, para evaluar $\Lambda^T \Upsilon \tilde{w}$ tan sólo es necesario evaluar [2.3.19], ya que la matriz A ya se calculó para obtener [2.3.11].

Una vez descrita la evaluación de los elementos de $\mathbf{T}\tilde{\mathbf{w}}$, $\mathbf{\Lambda}^T\mathbf{T}\tilde{\mathbf{w}}$ y $\mathbf{\Lambda}^T\mathbf{\Lambda}$, el cálculo de la *FVE* [2.2.42] puede llevarse a cabo mediante la siguiente secuencia de operaciones (algoritmo [FVE']):

- [FVE'.1] Realizar los pasos [NH.1]-[NH.5] del algoritmo [NH] [Hall y Nicholls (1980)].
- [FVE'.2] Evaluar los nm elementos no nulos (vector η) de $\mathbf{T}\tilde{\mathbf{w}}$ según [2.3.2].
- [FVE'.3] Evaluar $\mathbf{\Lambda}^T\mathbf{\Lambda}$ según [2.3.11].
- [FVE'.4] Evaluar $\mathbf{\Lambda}^T\mathbf{T}\tilde{\mathbf{w}}$ según [2.3.20].
- [FVE'.5] Obtener la descomposición de Cholesky de $(\mathbf{\Lambda}^T\mathbf{\Lambda})=\mathbf{L}\mathbf{L}^T$ y evaluar $|\mathbf{\Lambda}^T\mathbf{\Lambda}|=|\mathbf{L}|^2$.
- [FVE'.6] Resolver el sistema triangular $\mathbf{L}\lambda=(\mathbf{\Lambda}^T\mathbf{T}\tilde{\mathbf{w}})$.
- [FVE'.7] Evaluar $(\mathbf{T}\tilde{\mathbf{w}})^T(\mathbf{T}\tilde{\mathbf{w}})=\eta^T\eta$ y $(\mathbf{\Lambda}^T\mathbf{T}\tilde{\mathbf{w}})^T(\mathbf{\Lambda}^T\mathbf{\Lambda})^{-1}(\mathbf{\Lambda}^T\mathbf{T}\tilde{\mathbf{w}})=\lambda^T\lambda$.

Respecto a [FVE'.5], $|\mathbf{\Lambda}^T\mathbf{\Lambda}|=|\mathbf{L}|^2$ (el producto de los elementos de la diagonal principal de \mathbf{L} al cuadrado). Por otro lado, la estrategia en [FVE'.6]-[FVE'.7] permite expresar la forma cuadrática [2.3.1] como la diferencia entre dos sumas de cuadrados (*vid.* Capítulo 3). Por último, análogamente a [NH.5], que proporciona una verificación suficiente de la no estacionariedad del proceso, [FVE'.3] lo proporciona respecto a la no invertibilidad, ya que la naturaleza recursiva de las expresiones [2.3.8]-[2.3.10] hace que estos cálculos resulten explosivos cuando el proceso no es invertible y n es lo suficientemente grande.

Es importante señalar que, mientras [FVE'.2] es equivalente al procedimiento de Hall y Nicholls (1980) para evaluar $(\mathbf{I}_n \otimes \mathbf{R})\mathbf{K}\tilde{\mathbf{w}}$, los pasos [FVE'.3]-[FVE'.7] son nuevos respecto a la literatura existente. Además, aunque el procedimiento [FVE'] resulta satisfactorio en la mayor parte de los casos encontrados en la práctica econométrica (*vid.* sección 2.4), la eficiencia computacional de este procedimiento puede mejorarse considerablemente cuando $g=\text{máximo}\{p,q\}$ es sensiblemente inferior a $p+q$ (cuando p y q son ambos elevados y, en cualquier caso, cuando $p \neq 0$ y $q \neq 0$ simultáneamente).

La clave para mejorar la eficiencia del algoritmo [FVE'] en estos casos, consiste en reemplazar la descomposición de Cholesky de las matrices de orden $(p+q)m \times (p+q)m$ $\mathbf{\Omega}$ y $\mathbf{\Lambda}^T\mathbf{\Lambda}$ (pasos [FVE'.1] y [FVE'.5]) por la de dos matrices de orden tan sólo $gm \times gm$. Para ello, es suficiente tener en cuenta que, en la expresión [2.2.48] para la forma cuadrática de la *FVE* [2.2.42], por la definición de $\mathbf{\Lambda}$ en [2.2.33]:

$$\mathbf{\Lambda}(\mathbf{\Lambda}^T\mathbf{\Lambda})^{-1}\mathbf{\Lambda}^T = \begin{pmatrix} (\mathbf{\Lambda}^T\mathbf{\Lambda})^{-1} & (\mathbf{\Lambda}^T\mathbf{\Lambda})^{-1}\mathbf{T}^{-1}\mathbf{Z}^T(\mathbf{I}_n \otimes \mathbf{R}^T) \\ (\mathbf{I}_n \otimes \mathbf{R})\mathbf{Z}\mathbf{T}^{-1}(\mathbf{\Lambda}^T\mathbf{\Lambda})^{-1} & (\mathbf{I}_n \otimes \mathbf{R})\mathbf{Z}\mathbf{T}^{-1}(\mathbf{\Lambda}^T\mathbf{\Lambda})^{-1}\mathbf{T}^{-1}\mathbf{Z}^T(\mathbf{I}_n \otimes \mathbf{R}^T) \end{pmatrix} \quad [2.3.21]$$

Además, ya que $\Upsilon\tilde{w} = (\mathbf{0}^T, \eta^T)^T$ con:

$$\eta = (\mathbf{I}_n \otimes \mathbf{R}) \hat{a}_0 \quad [2.3.22]$$

$$\hat{a}_0 = E[\mathbf{a} | \mathbf{w}, \mathbf{u}_* = \mathbf{0}] = \mathbf{K}\tilde{w} \quad [2.3.23]$$

(*vid.* [2.3.2]), la forma cuadrática [2.2.48] puede escribirse como:

$$(\Upsilon\tilde{w} + \Lambda\hat{e}_*)^T (\Upsilon\tilde{w} + \Lambda\hat{e}_*) = \eta^T \eta - \eta^T (\mathbf{I}_n \otimes \mathbf{R}) \mathbf{Z} \mathbf{T}^{-1} (\Lambda^T \Lambda)^{-1} \mathbf{T}^{-1T} \mathbf{Z}^T (\mathbf{I}_n \otimes \mathbf{R}^T) \eta \quad [2.3.24]$$

Pero teniendo en cuenta [2.3.3], [2.3.6] y [2.2.23]:

$$\mathbf{T}^T \Lambda^T \Lambda \mathbf{T} = \Omega^{-1} + \mathbf{V}_1^T \mathbf{H}^T \mathbf{H} \mathbf{V}_1 \quad [2.3.25]$$

y entonces:

$$\mathbf{T}^{-1} (\Lambda^T \Lambda)^{-1} \mathbf{T}^{-1T} = [\Omega^{-1} + \mathbf{V}_1^T \mathbf{H}^T \mathbf{H} \mathbf{V}_1]^{-1} \quad [2.3.26]$$

por lo que [2.3.24] puede escribirse como:

$$(\Upsilon\tilde{w} + \Lambda\hat{e}_*)^T (\Upsilon\tilde{w} + \Lambda\hat{e}_*) = \eta^T \eta - \eta^T (\mathbf{I}_n \otimes \mathbf{R}) \mathbf{Z} [\Omega^{-1} + \mathbf{V}_1^T \mathbf{H}^T \mathbf{H} \mathbf{V}_1]^{-1} \mathbf{Z}^T (\mathbf{I}_n \otimes \mathbf{R}^T) \eta \quad [2.3.27]$$

y teniendo en cuenta [2.2.18] y [2.3.17]:

$$(\Upsilon\tilde{w} + \Lambda\hat{e}_*)^T (\Upsilon\tilde{w} + \Lambda\hat{e}_*) = \eta^T \eta - \mathbf{h}^T \mathbf{V} [\Omega^{-1} + \mathbf{V}_1^T \mathbf{H}^T \mathbf{H} \mathbf{V}_1]^{-1} \mathbf{V}^T \mathbf{h} \quad [2.3.28]$$

o bien:

$$(\Upsilon\tilde{w} + \Lambda\hat{e}_*)^T (\Upsilon\tilde{w} + \Lambda\hat{e}_*) = \eta^T \eta - \tilde{\mathbf{h}}^T \mathbf{V}_1 [\Omega^{-1} + \mathbf{V}_1^T \mathbf{H}^T \mathbf{H} \mathbf{V}_1]^{-1} \mathbf{V}_1^T \tilde{\mathbf{h}} \quad [2.3.29]$$

donde (*vid.* Apéndice A.1):

$$\mathbf{V}_1 [\Omega^{-1} + \mathbf{V}_1^T \mathbf{H}^T \mathbf{H} \mathbf{V}_1]^{-1} \mathbf{V}_1^T = [(\mathbf{V}_1 \Omega \mathbf{V}_1^T)^{-1} + \mathbf{H}^T \mathbf{H}]^{-1} \quad [2.3.30]$$

por lo que [2.3.29] puede escribirse también como:

$$(\Upsilon\tilde{w} + \Lambda\hat{e}_*)^T (\Upsilon\tilde{w} + \Lambda\hat{e}_*) = \eta^T \eta - \tilde{\mathbf{h}}^T [(\mathbf{V}_1 \Omega \mathbf{V}_1^T)^{-1} + \mathbf{H}^T \mathbf{H}]^{-1} \tilde{\mathbf{h}} \quad [2.3.31]$$

Nótese que para evaluar [2.3.31] sólo es necesario obtener la descomposición de Cholesky de las matrices de orden $gm \times gm$ $(V_1 \Omega V_1^T)$ y $[(V_1 \Omega V_1^T)^{-1} + H^T H]$. Estas operaciones proporcionan simultáneamente la forma de evaluar el determinante $|\Lambda^T \Lambda|$, que puede escribirse, por [2.3.25], como:

$$|\Lambda^T \Lambda| = |\Omega| \times |\Omega^{-1} + V_1^T H^T H V_1| = |V_1 \Omega V_1^T| \times |(V_1 \Omega V_1^T)^{-1} + H^T H| \quad [2.3.32]$$

No obstante, las expresiones [2.3.31] y [2.3.32] presentan un ligero inconveniente desde el punto de vista numérico, que consiste en la necesidad de calcular *explícitamente* la inversa de la matriz $(V_1 \Omega V_1^T)$. En muchos casos, esto no supone ningún problema, pero en otros, la precisión de los cálculos puede verse seriamente afectada por esta circunstancia. Para evitar dicha inversión, es suficiente tener presente que, si definimos M ($gm \times gm$) como el factor de Cholesky de $(V_1 \Omega V_1^T)$:

$$(V_1 \Omega V_1^T) = M M^T \Leftrightarrow M^T (V_1 \Omega V_1^T)^{-1} M = I_{gm} \quad [2.3.33]$$

entonces, es evidente que:

$$\begin{aligned} [(V_1 \Omega V_1^T)^{-1} + H^T H] &= M^{-1T} M^T [(V_1 \Omega V_1^T)^{-1} + H^T H] M M^{-1} \\ &= M^{-1T} [I_{gm} + M^T H^T H M] M^{-1} \end{aligned} \quad [2.3.34]$$

por lo que:

$$[(V_1 \Omega V_1^T)^{-1} + H^T H]^{-1} = M [I_{gm} + M^T H^T H M]^{-1} M^T \quad [2.3.35]$$

y entonces, la forma cuadrática [2.3.31] puede escribirse definitivamente como:

$$(\Upsilon \tilde{w} + \Lambda \hat{e}_*)^T (\Upsilon \tilde{w} + \Lambda \hat{e}_*) = \eta^T \eta - (M^T \tilde{h})^T [I_{gm} + M^T H^T H M]^{-1} (M^T \tilde{h})$$

[2.3.36]

Por último, a la vista de [2.3.32]-[2.3.34], resulta inmediato comprobar que el determinante $|\Lambda^T \Lambda|$ puede escribirse como:

$$|\Lambda^T \Lambda| = |I_{gm} + M^T H^T H M|$$

[2.3.37]

Por tanto, para evaluar la *FVE* de un proceso ARMA estacionario, basta con introducir las expresiones [2.3.36] y [2.3.37] en la formulación [2.2.42] de Nicholls y Hall (1979). Ya que tanto el cálculo de los g vectores \mathbf{h}_i , necesarios en [2.3.36], como el cálculo de la matriz $\mathbf{H}^T\mathbf{H}$, necesaria en [2.3.36] y [2.3.37], están descritos en [2.3.19] y [2.3.9]-[2.3.10], respectivamente, tan sólo falta evaluar los $gm(gm+1)/2$ elementos distintos de la matriz $(\mathbf{V}_1\boldsymbol{\Omega}\mathbf{V}_1^T)$, operación que puede llevarse a cabo, para $i=1,2,\dots,g, j=1,2,\dots,i$, como sigue:

$$(\mathbf{V}_1\boldsymbol{\Omega}\mathbf{V}_1^T)_{ij} = \sum_{k=0}^{p-i} \boldsymbol{\Phi}_{p-k}(\boldsymbol{\Omega}\mathbf{V}_1^T)_{k+i,j} - \sum_{k=0}^{q-i} \boldsymbol{\Theta}_{q-k}(\boldsymbol{\Omega}\mathbf{V}_1^T)_{k+p+i,j} \quad [2.3.38]$$

donde, para $j=1,2,\dots,g$:

$$(\boldsymbol{\Omega}\mathbf{V}_1^T)_{ij} = \sum_{k=j-i}^{p-i} \boldsymbol{\Gamma}(k)\boldsymbol{\Phi}_{p-k-i+j}^T - \sum_{k=j-i}^{q-i} \boldsymbol{\Gamma}_{wa}(-q+p+k)\boldsymbol{\Theta}_{q-k-i+j}^T, \quad i=1,\dots,p \quad [2.3.39]$$

$$(\boldsymbol{\Omega}\mathbf{V}_1^T)_{ij} = \sum_{k=p+j-i}^{2p-i} \boldsymbol{\Gamma}_{wa}(-q+p-k)^T\boldsymbol{\Phi}_{2p-k-i+j}^T - \boldsymbol{Q}\boldsymbol{\Theta}_{q+p-i+j}^T, \quad i=p+1,\dots,p+q \quad [2.3.40]$$

con $\boldsymbol{\Gamma}(k)=\boldsymbol{\Gamma}(-k)^T$ para $k<0$, $\boldsymbol{\Gamma}_{wa}(k)=\mathbf{0}$ para $k>0$ y $\boldsymbol{\Theta}_i=\mathbf{0}$ para $i>q$. En definitiva, el algoritmo propuesto para evaluar eficientemente la *FVE*, puede resumirse en los siguientes pasos (algoritmo [FVE]):

- [FVE.1] Realizar los pasos [NH.1]-[NH.5] del algoritmo [NH] (sin calcular \mathbf{T}^{-1}).
- [FVE.2] Evaluar los nm elementos no nulos (vector $\boldsymbol{\eta}$) de $\mathbf{T}\tilde{\mathbf{w}}$ según [2.3.2].
- [FVE.3] Evaluar $\mathbf{h}_i, i=1,2,\dots,g$, según [2.3.19].
- [FVE.4] Evaluar $\mathbf{H}^T\mathbf{H}$ según [2.3.9] y [2.3.10].
- [FVE.5] Evaluar $(\mathbf{V}_1\boldsymbol{\Omega}\mathbf{V}_1^T)$ según [2.3.38]-[2.3.40].
- [FVE.6] Obtener \mathbf{M} tal que $(\mathbf{V}_1\boldsymbol{\Omega}\mathbf{V}_1^T)=\mathbf{M}\mathbf{M}^T$.
- [FVE.7] Obtener \mathbf{L} tal que $[\mathbf{I}_{gm}+\mathbf{M}^T\mathbf{H}^T\mathbf{H}\mathbf{M}]=\mathbf{L}\mathbf{L}^T$ y evaluar $|\boldsymbol{\Lambda}^T\boldsymbol{\Lambda}|=|\mathbf{I}_{gm}+\mathbf{M}^T\mathbf{H}^T\mathbf{H}\mathbf{M}|=|\mathbf{L}|^2$.
- [FVE.8] Resolver el sistema triangular $\mathbf{L}\boldsymbol{\lambda}=(\mathbf{M}^T\tilde{\mathbf{h}})$.
- [FVE.9] Evaluar la forma cuadrática [2.3.36] como $(\boldsymbol{\eta}^T\boldsymbol{\eta} - \boldsymbol{\lambda}^T\boldsymbol{\lambda})$.

Antes de comentar algunos detalles computacionales del algoritmo [FVE], es importante tener en cuenta las siguientes observaciones:

- [1] La estrategia empleada para obtener las expresiones [2.3.31] y [2.3.32], consiste básicamente en la extensión al contexto multivariante de algunas de las ideas sugeridas por Ljung y Box (1979). En concreto, cuando $m = 1$, las expresiones [2.3.27] y [2.3.32] se reducen a las expresiones (2.6) y (2.4), respectivamente, de Ljung y Box (1979). Por tanto, ya que dichas expresiones se han obtenido a partir de la formulación [2.2.42] de Nicholls y Hall (1979), constituyen una demostración implícita de la equivalencia matemática entre ambos trabajos.
- [2] No obstante, la estrategia que evita invertir explícitamente la matriz $(V_1 \Omega V_1^T)$ y que conduce a las expresiones [2.3.36] y [2.3.37], no se encuentra sugerida en ninguna referencia (hasta donde conocemos). Además de la mejora, en términos de precisión numérica, que supone dicha estrategia frente a una extensión estricta del trabajo de Ljung y Box (1979), también proporciona una mejora en términos de velocidad de cálculo, ya que para invertir la matriz $(V_1 \Omega V_1^T)$ son necesarias $n_1 = g^2 m^2 (gm + 2)$ multiplicaciones y divisiones, mientras que para evaluar las expresiones $M^T H^T H M$ y $M^T \tilde{h}$ sólo son necesarias $n_2 = g^2 m^2 (gm + 1) + gm(gm + 1)/2$ multiplicaciones (nótese que $n_1 - n_2 = gm(gm - 1)/2 > 0$, excepto cuando $m = g = 1$).
- [3] La estrategia que se utiliza para evaluar la forma cuadrática [2.3.36] como la diferencia entre dos sumas de cuadrados (pasos [FVE.8]-[FVE.9]), permite maximizar la *FVE* utilizando tanto algoritmos de propósito general (tipo *cuasi-Newton*; *vid.* Capítulo 3) como algoritmos específicos para sumas de cuadrados (tipo *Gauss-Newton*; *vid.* Capítulo 3), de manera que el algoritmo [FVE] puede adaptarse a los métodos más eficientes de optimización numérica disponibles en cualquier librería científica.
- [4] El vector de residuos correspondientes a la muestra considerada y a los valores numéricos de los parámetros del proceso ARMA, puede obtenerse directamente a partir de las operaciones descritas para evaluar [2.3.36] y [2.3.37]. En concreto, teniendo en cuenta [2.2.16], $\hat{a} = K\tilde{w} + Z\hat{u}_*$, donde $K\tilde{w} = \hat{a}_0$ (ya calculado; *vid.* [2.3.2]) y, por [2.2.24]-[2.2.25], $\hat{u}_* = T^{-1}\hat{e}_*$, donde $\hat{e}_* = E[e_* | w]$ está definido en [2.2.36]. Por lo tanto, $\hat{u}_* = -T^{-1}(\Lambda^T \Lambda)^{-1} \Lambda^T \Upsilon \tilde{w}$ y, teniendo en cuenta [2.3.15], [2.3.16] y [2.3.26], resulta finalmente que:

$$\hat{u}_* = [\Omega^{-1} + V_1^T H^T H V_1]^{-1} V_1^T h \quad [2.3.41]$$

A partir de esta expresión, es inmediato comprobar que:

$$\hat{\mathbf{a}} = \hat{\mathbf{a}}_0 - \mathbf{D}_{\Theta,n}^{-1} \begin{bmatrix} \mathbf{M}[\mathbf{I}_{gm} + \mathbf{M}^T \mathbf{H}^T \mathbf{H} \mathbf{M}]^{-1} \mathbf{M}^T \tilde{\mathbf{h}} \\ \mathbf{0}_{(n-g)m \times 1} \end{bmatrix} \quad [2.3.42]$$

Por tanto, a partir de cantidades ya evaluadas en el algoritmo [FVE], el vector de residuos estimados $\hat{\mathbf{a}}$ puede calcularse mediante la siguiente secuencia de operaciones (algoritmo [CRES]):

[CRES.1] Resolver el sistema triangular $\mathbf{L}^T \mathbf{c} = \lambda$ y evaluar el vector $\mathbf{d} = \mathbf{M} \mathbf{c}$.

[CRES.2] Evaluar el vector $\mathbf{r} = \mathbf{D}_{\Theta,n}^{-1} (\mathbf{d}^T, \mathbf{0}^T)^T$ y calcular $\hat{\mathbf{a}} = \hat{\mathbf{a}}_0 - \mathbf{r}$.

La evaluación de los n vectores r_i que componen el vector \mathbf{r} de [CRES.2], puede llevarse a cabo, teniendo en cuenta la estructura de la matriz $\mathbf{D}_{\Theta,n}^{-1}$ (vid. [2.1.23]), de la siguiente forma:

$$r_i = \sum_{j=1}^i \Xi_{i-j} d_j, \quad i=1,2,\dots,n \quad [2.3.43]$$

con $d_j = \mathbf{0}$ para $j > g$. El vector de residuos así calculado puede emplearse para diagnosticar el modelo estimado y para generar previsiones con dicho modelo.

- [5] Los pasos [FVE.6] y [FVE.4] proporcionan una comprobación automática de la estacionariedad y la invertibilidad del proceso considerado, ya que (i) la matriz $(\mathbf{V}_1 \boldsymbol{\Omega} \mathbf{V}_1^T)$ será definida positiva (condición necesaria y suficiente para la existencia de \mathbf{M}) sí y sólo sí lo es $\boldsymbol{\Omega}$ (condición necesaria para que el proceso sea estacionario y suficiente cuando $q=0$) y (ii) la recursividad de [2.3.8] hace que su cálculo resulte explosivo cuando el proceso no es invertible (vid. apartado 2.3.2).
- [6] Aunque el procedimiento descrito en el apartado 2.2.1 es válido para calcular las autocovarianzas teóricas (necesarias en [FVE.1]), podría incorporarse al algoritmo [FVE] el procedimiento más eficiente de Kohn y Ansley (1982). Evidentemente, este procedimiento también podría incorporarse a cualquier otro algoritmo para evaluar la FVE.

2.3.2. Algunos aspectos computacionales

La principal ventaja del procedimiento descrito en el apartado 2.2.3 [Shea (1989)], consiste en la posibilidad de adoptar la estrategia conocida como "switching to quick recursions" (SQR en adelante) en

algún momento del proceso de evaluación de la función de verosimilitud. Como escribe Shea (1989, pp. 169): "*This 'quick recursions' property gives the Kalman filter a major computational advantage over other methods*". La idea original de esta estrategia, en el contexto de procesos univariantes, puede encontrarse en Gardner, Harvey y Phillips (1980) y ha sido aplicada (dentro de un algoritmo más eficiente) por Melard (1984), también para procesos univariantes.

Una estrategia similar a *SQR* puede aplicarse dentro del algoritmo [FVE]. En concreto, si el proceso multivariante considerado es invertible, no suele ser necesario calcular las matrices Ξ_i según [2.3.8] hasta $i=n-1$ ya que, a medida que i crece, $\Xi_i \rightarrow \mathbf{0}$ tanto más rápidamente cuanto mayores sean los módulos de las raíces del polinomio característico [2.1.7]. Evidentemente, para un proceso AR puro ($q=0$), esta convergencia ocurre exactamente cuando $i=1$. Además, una verificación de la invertibilidad del proceso considerado consiste únicamente en comprobar que esta convergencia ocurre (tarea trivial en comparación con el cálculo explícito de las raíces de [2.1.7]). Esta propiedad (al igual que *SQR*) supone un importante ahorro en el número de operaciones requeridas para evaluar la función de verosimilitud (sobre todo, cuando n es grande; *vid.* [2.3.8]-[2.3.10] y [2.3.19], operaciones contenidas en [FVE.3] y [FVE.4]). El valor de la función de verosimilitud, calculado haciendo uso de esta propiedad, puede considerarse como una *aproximación* al valor de la *FVE*; por tanto, podemos hablar de una función de verosimilitud *aproximada* (*FVA* en adelante), en contraposición a la *FVE*, para cuyo cálculo no se hace uso de la propiedad de convergencia de las Ξ_i (excepto, lógicamente, para procesos AR puros, ya que en este caso los valores de la *FVE* y de la *FVA* coinciden).

En la práctica, podría pensarse que las matrices Ξ_i han convergido a $\mathbf{0}$ cuando se cumple la siguiente condición:

$$\left[\sum_{i=1}^m \sum_{j=1}^m |\Xi_{i*}(i,j)| \right] < \delta \quad [2.3.44]$$

No obstante, para evitar posibles complicaciones en modelos con algunas matrices $\Theta_i = \mathbf{0}$ para $i < q$ (en concreto, modelos estacionales), una vez satisfecha la condición [2.3.44], resulta conveniente calcular las q matrices Ξ_i siguientes a Ξ_{i*} , con el fin de asegurar que el cumplimiento de [2.3.44] se debe efectivamente a que las Ξ_i han convergido y no a la existencia de alguna $\Theta_i = \mathbf{0}$. En concreto, para determinar correctamente el índice (r^*) a partir del cual las Ξ_i ($i > r^*$) se consideran $\mathbf{0}$, puede utilizarse el procedimiento descrito en el Apéndice A.2 (algoritmo CXI).

En estas circunstancias, para $i > r^*$ puede considerarse que $\Xi_i = \mathbf{0}$ y utilizarse este hecho para

evaluar las expresiones [2.3.9], [2.3.10], [2.3.19] y, en su caso, [2.3.43]. El criterio de convergencia (representado por δ), puede hacerse lo suficientemente estricto (pequeño) como para que la diferencia entre los valores de la *FVE* y de la *FVA* sea despreciable (en concreto, cuando $\delta \leq 0$ los valores de la *FVE* y de la *FVA* coinciden). Para ilustrar todas estas cuestiones, consideremos el siguiente proceso bivariante:

$$\begin{bmatrix} 1 - \phi B^S & 0 \\ 0 & 1 - \phi B^S \end{bmatrix} \begin{bmatrix} w_{1t} \\ w_{2t} \end{bmatrix} = \begin{bmatrix} 1 - \theta B^S & 0 \\ 0 & 1 - \theta B^S \end{bmatrix} \begin{bmatrix} a_{1t} \\ a_{2t} \end{bmatrix} \quad [2.3.45]$$

En la Tabla 2.3.1, se presentan los resultados de la evaluación de la función de verosimilitud del proceso [2.3.45] (con $n=300$), para distintos valores de S , de ϕ y de θ . Nótese que para $\phi=0.999$ y $\theta=0.998$, el proceso es casi no estacionario y no invertible, respectivamente. En esta tabla también se presentan el índice r^* (obtenido para $\delta=10^{-3}$), a partir del cual se considera que $\Xi_i = \mathbf{0}$ ($i > r^*$) y el cociente entre el número de operaciones (multiplicaciones, divisiones y raíces cuadradas) requeridas para evaluar la *FVE* (N_E) y la *FVA* (N_A).

		$\phi = 0.5 ; \theta = 0.3$	$\phi = 0.999 ; \theta = 0.3$	$\phi = 0.5 ; \theta = 0.998$	$\phi = 0.999 ; \theta = 0.998$
ARMA(1,1) ($S = 1$)	<i>-log FVE</i>	1566.46949880	1036.08930170	104757.50462232	2201.89340436
	<i>-log FVA</i>	1566.46950727	1036.08913686	104757.50462232	2201.89340436
	r^*	6	6	299	299
	N_E/N_A	2.82	2.82	1.00	1.00
ARMA(1,1) ₄ ($S = 4$)	<i>-log FVE</i>	1895.68330334	1980.77848932	9274.77420320	2209.18545073
	<i>-log FVA</i>	1895.68327167	1980.77285648	9274.77420320	2209.18545073
	r^*	24	24	296	296
	N_E/N_A	2.54	2.54	1.00	1.00
ARMA(1,1) ₁₂ ($S = 12$)	<i>-log FVE</i>	2226.74584052	2930.91385243	2807.83186866	2212.02468385
	<i>-log FVA</i>	2226.74657427	2930.92252921	2807.83186866	2212.02468385
	r^*	72	72	288	288
	N_E/N_A	1.60	1.60	1.01	1.01

TABLA 2.3.1. Comparación en términos de precisión y carga computacional entre la evaluación exacta (*FVE*) y aproximada (*FVA*) de la función de verosimilitud del proceso [2.3.45] ($Q_{11}=Q_{21}=Q_{12}=1.0$, $Q_{22}=2.0$, $\sigma^2=1.0$).

Estos resultados sugieren las siguientes conclusiones [vid. sección 2.4]: (i) la evaluación de la *FVA* es extremadamente precisa en relación a la evaluación de la *FVE*; (ii) cuando el proceso considerado es claramente invertible, la ganancia en términos de tiempo de cálculo por evaluar la *FVA* en vez de la *FVE* es considerable; (iii) cuando el proceso es casi no invertible, evaluar la *FVA* es prácticamente equivalente a evaluar la *FVE*, ya que las matrices Ξ_i convergen a $\mathbf{0}$ muy lentamente y, por ello, r^* es prácticamente igual a $n-1$ (la discrepancia entre r^* y $n-1$ que ocurre en las dos últimas columnas de los procesos ARMA(1,1)₄ y ARMA(1,1)₁₂, se debe a la presencia de matrices $\Theta_i = \mathbf{0}$ para $i < 4$ e $i < 12$, respectivamente; no obstante, como debe ser, los valores calculados de la *FVA* y de la *FVE* son idénticos).

Finalmente, para comprobar la invertibilidad del proceso considerado, es suficiente comprobar si el cálculo de las matrices Ξ_i resulta convergente o explosivo. En concreto, podemos considerar que el proceso *no* es invertible cuando:

$$\left[\sum_{i=1}^m \sum_{j=1}^m |\Xi_h(i,j)| \right] > \sum_{k=1}^{\max(h,q)} \left[\sum_{i=1}^m \sum_{j=1}^m |\Xi_k(i,j)| \right] \quad [2.3.46]$$

para al menos un $h < n-1$. En general, cuando el proceso considerado *no* es invertible, la ocurrencia de la condición anterior tiene lugar para h ligeramente mayor que q , por lo que la detección de no invertibilidad suele ser casi inmediata (el procedimiento **CXI** del Apéndice A.2 también contiene esta comprobación de invertibilidad).

Un segundo aspecto importante, desde el punto de vista numérico, consiste en la caracterización de la matriz $(V_1 \Omega V_1^T)$ como no definida positiva, indicando esta situación la no estacionariedad del proceso considerado. Es importante subrayar que los problemas planteados por la no estacionariedad son bastante distintos, al menos desde el punto de vista numérico, de los planteados por la no invertibilidad. En concreto, mientras que el hecho de que $(V_1 \Omega V_1^T)$ no sea definida positiva implica la imposibilidad de realizar el paso **[FVE.6]** (y, por tanto, de evaluar la *FVE*), el hecho de que el proceso sea no invertible tan sólo ocasiona problemas numéricos (*overflow*) cuando n es lo suficientemente grande y el módulo de alguna de las raíces de [2.1.7] es claramente menor que la unidad (en particular, no existe ningún problema para evaluar la *FVE* en la frontera de la región de invertibilidad). Por ello, la caracterización de la matriz $(V_1 \Omega V_1^T)$ requiere especial atención (nótese, por otro lado, que estas últimas observaciones sugieren que, a la hora de construir un modelo ARMA, la sobrediferenciación puede resultar menos *costosa* que la infradiferenciación de las series estocásticas).

El mecanismo que se ha utilizado para caracterizar la matriz $(V_1 \Omega V_1^T)$, consiste en una modificación del algoritmo para obtener la descomposición de Cholesky de una matriz cuadrada propuesto por Gill, Murray y Wright (1981) y desarrollado, en forma de pseudocódigo, por Dennis y Schnabel (1983). Básicamente, la modificación realizada sobre este algoritmo (*vid.* Apéndice A.2, procedimiento **CHOLDCMP**) consiste en forzar el carácter definido positivo de la matriz $(V_1 \Omega V_1^T)$, solamente cuando en la descomposición de Cholesky estándar [*vid.* Martin, Peters y Wilkinson (1965)] se detecta que $(V_1 \Omega V_1^T)$ no es definida positiva debido a errores numéricos. Dichos errores numéricos surgen cuando (i) el proceso es casi no estacionario y/o (ii) el orden de magnitud de los elementos de la matriz \mathcal{Q} (que forma parte de Ω) es pequeño. En estos casos, la descomposición de Cholesky estándar puede concluir

(erróneamente) que $(V_1 \Omega V_1^T)$ no es definida positiva.

El procedimiento **CHOLDCMP** también se emplea en [FVE.7], para obtener la descomposición de Cholesky de $[I_{gm} + M^T H^T H M]$ y para calcular su determinante. Respecto a este segundo cálculo, se ha evitado la posible presencia de problemas numéricos (*overflow* o *underflow*), almacenando dicho determinante en la forma $a2^b$ [vid. Martin y Wilkinson (1965); Ansley (1979); Melard (1984)]. La misma estrategia se ha seguido para calcular la descomposición de Cholesky de Q y su determinante en [FVE.1].

Un último aspecto computacionalmente relevante, consiste en la discusión de las necesidades de almacenamiento que presenta el algoritmo [FVE]. Dado que este punto se trata con detalle en el Apéndice A.2, baste señalar ahora que: (i) salvo para procesos MA puros, la cantidad máxima de almacenamiento requerida suele venir dictada por la necesaria para evaluar las autocovarianzas teóricas y (ii) a modo de ejemplo, resulta posible evaluar la FVE de procesos ARMA con $m=4$, $p=q=13$ y $n=300$ dentro de los límites reducidos de un ordenador personal (concretamente, todos los cálculos presentados en este trabajo, se han llevado a cabo en un *HP Vectra 386/25* operando bajo *MS-DOS 5.0*, lo que supone disponer de algo menos de 600 Kb de memoria RAM para código y datos).

2.3.3. Notas para la sección 2.3

[1] Dado que Hillmer y Tiao (1979) sólo obtienen la FVE para un MA(q) puro, nos limitaremos a exponer la equivalencia en este caso. Cuando $p=0$, $\Omega = (I_q \otimes Q)$ (vid. [2.2.43]-[2.2.47]). Por tanto, teniendo en cuenta [2.3.32], [2.3.6], [2.3.4] y [2.2.22], es evidente que:

$$|\Lambda^T \Lambda| = |Q|^q \times |(I_q \otimes Q^{-1}) + Z^T (I_q \otimes Q^{-1}) Z|$$

Por otro lado, sustituyendo [2.3.22]-[2.3.23] en [2.3.27], obtenemos que:

$$(\mathbf{T}\tilde{w} + \Lambda \hat{e}_*)^T (\mathbf{T}\tilde{w} + \Lambda \hat{e}_*) = \tilde{w}^T K^T \{ (I_n \otimes Q^{-1}) - (I_n \otimes Q^{-1}) Z [(I_q \otimes Q^{-1}) + Z^T (I_n \otimes Q^{-1}) Z]^{-1} Z^T (I_n \otimes Q^{-1}) \} K \tilde{w}$$

Finalmente, sustituyendo estas dos expresiones en la FVE [2.2.42], teniendo presente que para un MA(q) $Z = -D_{\theta,n}^{-1} G_{\theta,n}$ y $K = D_{\theta,n}^{-1}$, se obtiene la formulación propuesta por Hillmer y Tiao (1979).

2.4. Análisis comparativo de distintos procedimientos

Partiendo del planteamiento realizado en las secciones 2.2 y 2.3, puede concluirse que el algoritmo [FVE] tiene el potencial de ser computacionalmente más eficiente (en términos de velocidad de cálculo y necesidades de almacenamiento) que los algoritmos de Hall y Nicholls (1979), Hillmer y Tiao (1979) (para procesos MA puros) y la extensión multivariante del algoritmo de Ljung y Box (1979), puesto que el algoritmo [FVE] explota hasta sus últimas consecuencias la estructura de los factores que aparecen en la FVE [2.2.42]. En este sentido, puede decirse que los tres trabajos mencionados no explotan todas las posibilidades que parecen ofrecer, incluso ignorando los aspectos computacionales descritos en el apartado 2.3.2 (en particular, la discusión sobre la evaluación de la FVA).

Por este motivo y porque una comparación teórica con los algoritmos basados en representaciones en forma *EE* resulta complicada y confusa, se ha considerado conveniente comparar el algoritmo [FVE] sólo con el más relevante de los basados en alguna representación en forma *EE* [Shea (1987, 1989)], prestando especial atención a las cuestiones prácticas.

Para realizar la comparación empírica se han escrito programas de ordenador basados en la evaluación directa de la FVE (apartado 2.2.1), en el algoritmo de Hall y Nicholls (1980) (algoritmo [NH], apartado 2.2.2), en el algoritmo [FVE'] (apartado 2.3.1), en la extensión multivariante del algoritmo de Ljung y Box (1979) (apartado 2.3.1), en el algoritmo [FVE] (apartado 2.3.2) y en el algoritmo de Shea (1989) (apartado 2.2.3). Los resultados de distintas comparaciones confirman, lógicamente, las conclusiones expuestas más arriba.

Todos los programas mencionados se han escrito en *Pascal* utilizando precisión doble, se han compilado con *Turbo Pascal* 6.0 y se han ejecutado en un *HP Vectra* 386/25 con un coprocesador matemático *Intel* 387DX-25 (un equipo que, actualmente, puede considerarse estándar). Al escribir en *Pascal* el algoritmo de Shea (1989), publicado en *FORTTRAN*, se ha respetado absolutamente la estructura del código original, de forma que el programa resultante es al menos tan eficiente como el original (en algunos casos, al hacer uso de características propias de *Pascal*, incluso más).

Para comparar la eficiencia computacional (en términos de tiempo de cálculo) entre el procedimiento de Shea (1989) y el algoritmo [FVE], se ha evaluado la FVE de un amplio conjunto de procesos ARMA multivariantes (*vid.* Tabla 2.4.1) para distintos valores de m y de n .

M 1	AR(1)	M 9	AR(1) ₄	M 17	AR(1) × ARMA(1,1) ₄	M 25	MA(1) × MA(1) ₁₂
M 2	AR(2)	M 10	MA(1) ₄	M 18	AR(1) × AR(1) ₁₂	M 26	MA(1) × ARMA(1,1) ₁₂
M 3	MA(1)	M 11	ARMA(1,1) ₄	M 19	AR(1) × MA(1) ₁₂	M 27	ARMA(1,1) × AR(1) ₄
M 4	MA(2)	M 12	AR(1) ₁₂	M 20	AR(1) × ARMA(1,1) ₁₂	M 28	ARMA(1,1) × MA(1) ₄
M 5	ARMA(1,1)	M 13	MA(1) ₁₂	M 21	MA(1) × AR(1) ₄	M 29	ARMA(1,1) × ARMA(1,1) ₄
M 6	ARMA(2,1)	M 14	ARMA(1,1) ₁₂	M 22	MA(1) × MA(1) ₄	M 30	ARMA(1,1) × AR(1) ₁₂
M 7	ARMA(1,2)	M 15	AR(1) × AR(1) ₄	M 23	MA(1) × ARMA(1,1) ₄	M 31	ARMA(1,1) × MA(1) ₁₂
M 8	ARMA(2,2)	M 16	AR(1) × MA(1) ₄	M 24	MA(1) × AR(1) ₁₂	M 32	ARMA(1,1) × ARMA(1,1) ₁₂

TABLA 2.4.1. Modelos empleados para comparar la eficiencia computacional de los algoritmos de Shea (1989) y [FVE] ($m=2, 4$; $n=100, 200, 300$).

MODELOS	$m = 2$			$m = 4$		
	$n = 100$	$n = 200$	$n = 300$	$n = 100$	$n = 200$	$n = 300$
M 1	1.01 *	1.01 *	1.00 *	1.05 *	1.03 *	1.02 *
M 2	0.99	0.99	1.00 *	1.00 *	1.00 *	1.00 *
M 3	2.38 *	2.40 *	2.41 *	2.51 *	2.53 *	2.54 *
M 4	1.85 *	1.88 *	1.89 *	1.95 *	1.98 *	1.99 *
M 5	2.42 *	2.46 *	2.47 *	2.65 *	2.69 *	2.70 *
M 6	2.02 *	2.07 *	2.09 *	2.14 *	2.20 *	2.22 *
M 7	1.90 *	1.94 *	1.96 *	2.05 *	2.09 *	2.11 *
M 8	1.94 *	1.99 *	2.01 *	2.12 *	2.18 *	2.21 *
M 9	0.81	0.88	0.91	0.73	0.81	0.86
M 10	1.44 *	1.51 *	1.53 *	1.53 *	1.60 *	1.63 *
M 11	1.52 *	1.63 *	1.68 *	1.65 *	1.79 *	1.85 *
M 12	0.30	0.39	0.46	0.23	0.29	0.34
M 13	0.77	0.96	1.06 *	0.78	1.01 *	1.12 *
M 14	0.70	0.96	1.10 *	0.69	0.99	1.16 *
M 15	0.71	0.80	0.85	0.61	0.71	0.76
M 16	1.48 *	1.55 *	1.58 *	1.58 *	1.66 *	1.69 *
M 17	1.46 *	1.61 *	1.66 *	1.50 *	1.69 *	1.76 *
M 18	0.27	0.35	0.42	0.21	0.26	0.31
M 19	0.77	0.97	1.07 *	0.79	1.02 *	1.13 *
M 20	0.69	0.97	1.12 *	0.65	0.95	1.13 *
M 21	1.48 *	1.59 *	1.63 *	1.51 *	1.64 *	1.69 *
M 22	1.32 *	1.41 *	1.44 *	1.40 *	1.50 *	1.53 *
M 23	1.38 *	1.51 *	1.56 *	1.48 *	1.64 *	1.71 *
M 24	0.58	0.79	0.90	0.52	0.73	0.85
M 25	0.71	0.92	1.02 *	0.72	0.95	1.07 *
M 26	0.65	0.91	1.05 *	0.64	0.93	1.10 *
M 27	1.30 *	1.43 *	1.49 *	1.30 *	1.46 *	1.52 *
M 28	1.34 *	1.44 *	1.47 *	1.43 *	1.54 *	1.59 *
M 29	1.38 *	1.52 *	1.58 *	1.47 *	1.66 *	1.74 *
M 30	0.52	0.73	0.84	0.46	0.66	0.79
M 31	0.71	0.92	1.03 *	0.72	0.96	1.08 *
M 32	0.64	0.90	1.04 *	0.63	0.92	1.09 *

TABLA 2.4.2. Eficiencia computacional relativa de los algoritmos de Shea (1989) y [FVE] (un * indica cuándo [FVE] es más eficiente en términos de tiempo de cálculo).

Este conjunto de modelos es bastante representativo de los utilizados habitualmente en la práctica y contiene el suficiente número de ellos como para poder sacar conclusiones, respecto a la eficiencia de los dos algoritmos considerados, generalizables para cualesquiera valores de p y q . Con este propósito, la Tabla 2.4.2 contiene el cociente entre el número de operaciones (multiplicaciones, divisiones y raíces cuadradas) requeridas por el algoritmo de Shea (1989) y las requeridas por el algoritmo [FVE], para evaluar la FVE de los modelos considerados (por tanto, un cociente superior a la unidad implica que el algoritmo [FVE] es más eficiente, en términos de tiempo de cálculo, que el de Shea (1989), para las especificaciones de p , q , m y n correspondientes).

A la vista de estos resultados, pueden extraerse las siguientes conclusiones importantes: (i) la eficiencia relativa del algoritmo [FVE] aumenta con n (excepto en procesos AR de orden bajo) y con m (excepto si p es elevado y bastante mayor que q) y (ii) el algoritmo de Shea (1989) sólo es sensiblemente superior a [FVE] cuando p es elevado y bastante mayor que q , o cuando p y q son ambos elevados y la muestra considerada no es larga (en este segundo caso, las diferencias son pequeñas).

Por tanto, ya que en la práctica econométrica es poco realista suponer que a una serie corta se le puede ajustar un modelo de orden elevado, puede concluirse que el algoritmo [FVE] es más eficiente que el de Shea (1989) en todos los casos de interés práctico, excepto cuando p es elevado y bastante mayor que q . Para ilustrar esta conclusión, en la Tabla 2.4.3 se presenta el mismo tipo de información que en la Tabla 2.4.2, ampliada y reorganizada para contemplar la eficiencia relativa de ambos algoritmos cuando se aplican a modelos adecuados para datos anuales, trimestrales y mensuales, con tamaños muestrales equivalentes a disponer de 25 años de datos (para que los resultados variasen significativamente, debería contemplarse un número de años bastante inferior a 25, lo cual sería poco realista).

Para finalizar la comparación, desde el punto de vista computacional, entre los algoritmos de Shea (1989) y [FVE] puede señalarse que: (i) las necesidades de almacenamiento del algoritmo [FVE] son, con algunas excepciones, ligeramente superiores a las del algoritmo de Shea (1989) (*vid.* Apéndice A.2), aunque, en ambos casos, dichas necesidades son extremadamente pequeñas comparadas, por ejemplo, con las requeridas para evaluar la FVE directamente (apartado 2.2.1) o mediante el algoritmo de Hall y Nicholls (1989) (apartado 2.2.2); (ii) en todos los casos considerados, los resultados numéricos de los algoritmos de Shea (1989) y [FVE], coinciden hasta al menos la octava posición decimal significativa; (iii) a modo de curiosidad, para la evaluación de la FVE de todos los ejemplos considerados en Shea (1987, 1989), el algoritmo [FVE] requiere menos operaciones que el propuesto en dichos trabajos.

MODELOS	Modelos para datos anuales ($n = 25$)	
	$m = 2$	$m = 4$
M 1	1.04 *	1.14 *
M 2	0.97	1.00 *
M 3	2.24 *	2.39 *
M 4	1.68 *	1.78 *
M 5	2.25 *	2.46 *
M 6	1.76 *	1.84 *
M 7	1.69 *	1.81 *
M 8	1.68 *	1.81 *
MODELOS	Modelos para datos trimestrales ($n = 100$)	
	$m = 2$	$m = 4$
M 9	0.81	0.73
M 10	1.44 *	1.53 *
M 11	1.52 *	1.65 *
M 15	0.71	0.61
M 16	1.48 *	1.58 *
M 17	1.46 *	1.50 *
M 21	1.48 *	1.51 *
M 22	1.32 *	1.40 *
M 23	1.38 *	1.48 *
M 27	1.30 *	1.30 *
M 28	1.34 *	1.43 *
M 29	1.38 *	1.47 *
MODELOS	Modelos para datos mensuales ($n = 300$)	
	$m = 2$	$m = 4$
M 12	0.46	0.34
M 13	1.06 *	1.12 *
M 14	1.10 *	1.16 *
M 18	0.42	0.31
M 19	1.07 *	1.13 *
M 20	1.12 *	1.13 *
M 24	0.90	0.85
M 25	1.02 *	1.07 *
M 26	1.05 *	1.10 *
M 30	0.84	0.79
M 31	1.03 *	1.08 *
M 32	1.04 *	1.09 *

TABLA 2.4.3. Eficiencia computacional relativa de los algoritmos de Shea (1989) y [FVE] cuando se aplican a modelos para datos anuales, trimestrales y mensuales (un * indica cuándo [FVE] es más eficiente en términos de tiempo de cálculo).

Como se indicó en el apartado 2.3.2, una propiedad interesante del algoritmo [FVE] [compartida por el de Shea (1989)] consiste en la posibilidad de controlar el grado de precisión en la evaluación de la función de verosimilitud. Haciendo uso en el algoritmo [FVE] de la propiedad de convergencia de las matrices \tilde{E}_i [o de la convergencia de K_t , F_t y v_t en el de Shea (1989)], puede obtenerse una evaluación numéricamente precisa de la función de verosimilitud, tanto más cuanto más rígida sea la tolerancia

(valor del parámetro δ) empleada. Si, además, las raíces de [2.1.7] están claramente fuera del círculo unitario, la ganancia en tiempo de cálculo, utilizando cualquiera de los dos algoritmos, respecto a una evaluación exacta puede ser considerable. En este sentido, la Tabla 2.4.4 contiene la ganancia media en el número de operaciones y el error relativo medio en el valor del logaritmo de la función de verosimilitud, resultantes de la evaluación aproximada (con $\delta = 10^{-3}$) frente a la evaluación exacta de la función de verosimilitud de los modelos de la Tabla 2.4.1 (excluidos los procesos AR puros).

Los resultados se han tabulado para las aproximaciones implícitas en los dos algoritmos considerados (*vid.* apartados 2.2.3 y 2.3.2 para Shea (1989) y [FVE], respectivamente). Los valores numéricos de los parámetros MA se han escogido de tal forma que los procesos resultantes son claramente invertibles. Evidentemente, para procesos cerca de la no invertibilidad la ganancia en el número de operaciones resultaría inferior a las presentadas en la tabla, ya que, en este caso, la convergencia de $\bar{\mathbf{z}}_i$, \mathbf{K}_t , \mathbf{F}_t y \mathbf{v}_t ocurre más lentamente.

		$m = 2$			$m = 4$		
		$n = 100$	$n = 200$	$n = 300$	$n = 100$	$n = 200$	$n = 300$
Shea (1989)	Ganancia media	3.80	4.76	5.22	5.07	6.89	7.91
	Error relativo medio	0.22×10^{-3}	0.23×10^{-3}	0.24×10^{-3}	0.28×10^{-3}	0.29×10^{-3}	0.29×10^{-3}
Algoritmo [FVE]	Ganancia media	1.83	2.29	2.56	2.04	2.76	3.22
	Error relativo medio	0.88×10^{-6}	0.45×10^{-6}	0.30×10^{-6}	0.41×10^{-6}	0.21×10^{-6}	0.14×10^{-6}

TABLA 2.4.4. Comparación entre las evaluaciones aproximada y exacta de la función de verosimilitud de los modelos de la Tabla 2.4.1 utilizando los algoritmos de Shea (1989) y [FVE].

Si bien la ganancia media en el número de operaciones es, en el contexto del algoritmo de Shea (1989), algo más del doble que en el contexto del algoritmo [FVE], el error relativo medio asociado a la evaluación aproximada es, en aquel caso, del orden de 1000 veces mayor que en el caso del algoritmo [FVE]. En ambos casos, la ganancia en tiempo de cálculo es grande y creciente con n y con m . No obstante, la precisión numérica de la evaluación aproximada en el caso del algoritmo [FVE] es considerable y aumenta con la dimensión del modelo (n y m), lo cual no parece ocurrir en el caso del algoritmo de Shea (1989). En definitiva, estas propiedades hacen de la FVA, implicada por cualquiera de los dos algoritmos, un interesante candidato como función objetivo en el proceso de estimación de los parámetros de cualquier modelo considerado (*vid.* Capítulo 3).

Para concluir el análisis comparativo de esta sección, es importante tener en cuenta, aparte de la eficiencia computacional, otras propiedades de los algoritmos considerados, que pueden hacer a uno de

ellos más interesante que los otros en determinados contextos. En concreto, creemos que el siguiente conjunto de propiedades debe tenerse en cuenta a la hora de comparar las prestaciones de distintos algoritmos para evaluar la función de verosimilitud de procesos ARMA (no necesariamente por orden de importancia): (i) que la evaluación sea *analíticamente* exacta; (ii) la posibilidad de realizar una evaluación *aproximada*, no consistente en ignorar ciertos términos de las expresiones analíticas exactas (*vid.* Capítulo 3), que sea numéricamente precisa y considerablemente más rápida que la evaluación exacta; (iii) su eficiencia computacional, en términos de tiempo de cálculo y necesidades de almacenamiento; (iv) su precisión numérica, sobre todo en situaciones mal condicionadas; (v) detección *automática* (como subproducto de ciertos cálculos) de situaciones de no estacionariedad o no invertibilidad, sin necesidad de resolver complicados polinomios; (vi) finalmente, la posibilidad de generar, a partir de cálculos ya efectuados para evaluar la función de verosimilitud, el vector de residuos *analíticamente exactos* correspondiente a los valores numéricos de los parámetros considerados.

En base a estos criterios, la Tabla 2.4.5 contiene un resumen comparativo de las prestaciones de todos los algoritmos considerados en este capítulo.

	Evaluación directa	Hall y Nicholls (1980)	Ljung y Box (1979)	Hillmer y Tiao (1979)	Shea (1987, 1989)	Algoritmo [FVE]
Evaluación exacta (<i>FVE</i>)	SI	SI	SI	NO	SI	SI
Evaluación aproximada (<i>FVA</i>)	NO	SI	SI	SI	SI	SI
Eficiencia computacional	NO	NO	NO	SI	SI	SI
Precisión numérica	SI	SI	NO	NO	SI	SI
Detección de no estacionariedad	SI	SI	SI	NO	SI	SI
Detección de no invertibilidad	NO	SI	SI	SI	NO	SI
Vector de residuos analíticamente exactos	NO	SI	SI	NO	NO	SI

TABLA 2.4.5. Comparación de los distintos algoritmos para evaluar la función de verosimilitud de procesos ARMA multivariantes.

El algoritmo [FVE] reúne, por tanto, un conjunto de propiedades, que se encuentran por separado en los algoritmos disponibles actualmente para evaluar la función de verosimilitud de procesos ARMA y no tiene ninguno de los inconvenientes que presentan dichos algoritmos. No obstante, aunque la validez *teórica* de este procedimiento queda fuera de duda, el análisis debe completarse con su puesta en práctica, frente a situaciones tanto simuladas como reales, en el contexto de la estimación de los parámetros de los modelos considerados. Este aspecto, junto con otros temas afines, se tratan en el capítulo siguiente.

CAPITULO 3

Estimación de procesos ARMA estacionarios por máxima verosimilitud

Una vez conocida la forma de evaluar la *FVE* de procesos ARMA estacionarios, es importante estudiar detalladamente la forma de maximizar dicha función (o alguna aproximación a ella), para obtener estimaciones de los parámetros que aparecen en los modelos considerados. Como es bien sabido, bajo ciertas condiciones generales, el criterio de estimación por máxima verosimilitud exacta (*MVE* en adelante) proporciona estimadores asintóticamente normales, consistentes [Whittle (1953)] y asintóticamente eficientes [Aigner (1971)]. Además, a partir del estudio de la estimación por *MVE*, pueden diseñarse otros mecanismos de estimación alternativos, con propiedades asintóticas equivalentes a las del estimador por *MVE*, que implican un coste computacional generalmente más reducido. Tanto el correcto funcionamiento en la práctica como el comportamiento en situaciones mal condicionadas (muestras pequeñas y parámetros redundantes o cerca de las fronteras de no estacionariedad y no invertibilidad) son aspectos que requieren un estudio cuidadoso para cualquiera de los métodos de estimación considerados. A continuación se resume brevemente el contenido de este capítulo:

Sección 3.1: Estimación por máxima verosimilitud exacta. En esta sección se describe un procedimiento computacionalmente eficiente para obtener los valores numéricos de los parámetros que maximizan la *FVE*, correspondientes a una muestra y una especificación dadas. También se ilustra la validez en la práctica del procedimiento descrito, a través de un conjunto de estimaciones en situaciones tanto simuladas como reales.

Sección 3.2: Otros procedimientos de estimación. A partir del criterio de estimación por máxima verosimilitud, se proponen y se estudian dos alternativas a su versión *exacta* (*MVE*): máxima verosimilitud *aproximada* (*MVA*) y máxima verosimilitud *condicional* (*MVC*). También se describe el criterio de estimación por *mínima suma de cuadrados* y sus posibles versiones.

Sección 3.3: Análisis comparativo de distintos procedimientos de estimación. Finalmente, se comparan las propiedades de las tres versiones de la estimación por máxima verosimilitud (descritas en las secciones anteriores) y del procedimiento ofrecido por *The S.C.A. Statistical System*.

3.1. Estimación por máxima verosimilitud exacta

La maximización numérica de la *FVE* para una muestra y una especificación dadas, requiere una formulación adecuada de la función a optimizar y un algoritmo eficiente de optimización numérica que se adapte a las características de esa formulación y, en la medida de lo posible, las aproveche. Básicamente, dichas características consisten en la necesidad de que los parámetros del modelo considerado cumplan determinadas condiciones, en especial las matrices \mathbf{Q} (*vid.* [2.1.11]) y Φ_i ($i=1,2,\dots,p$) (*vid.* [2.1.6] y apartado 2.3.2). Por ello, el procedimiento que se utilice para maximizar la *FVE* debe contemplar los requerimientos de dicha función para poder ser evaluada, así como las propiedades de un algoritmo de optimización que sea capaz de tener en cuenta dichos requerimientos, además de aprovechar la estructura de una formulación adecuada de la función objetivo.

Los trabajos sobre la optimización numérica de la *FVE* (o de alguna aproximación) de procesos ARMA son muy escasos. En el contexto univariante, Ansley (1979) y Kohn y Ansley (1985) demuestran que maximizar la *FVE* equivale a minimizar una suma de cuadrados y proponen para ello la utilización de algoritmos específicos para sumas de cuadrados (*vid.* Dennis y Schnabel (1983), cap. 10; Gill, Murray y Wright (1981), cap. 4). También en este contexto, Dent (1977) propone el empleo del algoritmo PRAXIS (un método de propósito general que no emplea derivadas; *vid.* Brent (1973), cap. 7) para maximizar la *FVE*. Esta es también la propuesta de Hall y Nicholls (1980) para procesos multivariantes.

En este contexto, Hillmer y Tiao (1979) proponen un procedimiento de estimación en dos etapas. En la primera etapa, sugieren estimar $\sigma^2\mathbf{Q}$ mediante las covarianzas muestrales de los residuos correspondientes a unas estimaciones (iniciales) de Φ_i ($i=1,2,\dots,p$), Θ_i ($i=1,2,\dots,q$) y μ . En la segunda etapa, proponen el empleo de un algoritmo para sumas de cuadrados para maximizar su aproximación a la *FVE*, con $\sigma^2\mathbf{Q}$ fija según el cálculo de la primera etapa. Las estimaciones de Φ_i ($i=1,2,\dots,p$), Θ_i ($i=1,2,\dots,q$) y μ así obtenidas, se utilizan en la primera etapa para obtener una nueva estimación de $\sigma^2\mathbf{Q}$ y el proceso se repite hasta que los elementos de $\sigma^2\mathbf{Q}$ convergen. Este procedimiento es el que ofrece *The S.C.A. Statistical System* para la estimación de modelos ARMA, tanto univariantes como multivariantes, incluso en su versión más reciente [*vid.* Liu y Hudak (1986, 1992)].

En todos estos trabajos, las propuestas mencionadas sólo se discuten brevemente y no se tratan, entre otros, temas tan fundamentales como las posibles estrategias a seguir cuando el algoritmo de optimización empleado genera un punto en el que no es posible evaluar la *FVE*. Esta cuestión se trata

detalladamente en Shea (1984, pp. 99-100) para procesos multivariantes. Una adaptación de las ideas de ese trabajo junto con ciertas extensiones, forman la base de esta sección.

3.1.1. Formulación de la función objetivo

Teniendo en cuenta las expresiones [2.3.36] y [2.3.37], la *FVE* [2.2.42] puede escribirse, en términos de los elementos evaluados por el algoritmo [FVE] (*vid.* apartado 2.3.1) como:

$$L(\Phi_i, \Theta_i, \mu, \sigma^2, \mathbf{Q} | \mathbf{w}) = (2\pi\sigma^2)^{-\frac{nm}{2}} \cdot |\mathbf{Q}|^{-\frac{n}{2}} \cdot |\mathbf{D}|^{-\frac{1}{2}} \cdot \exp\left[-\frac{1}{2\sigma^2}(\boldsymbol{\eta}^T \boldsymbol{\eta} - \boldsymbol{\lambda}^T \boldsymbol{\lambda})\right] \quad [3.1.1]$$

donde:

$$\mathbf{D} = \mathbf{I}_{mg} + \mathbf{M}^T \mathbf{H}^T \mathbf{H} \mathbf{M} \quad [3.1.2]$$

En [3.1.2], \mathbf{M} ($mg \times mg$) es el factor de Cholesky de $(\mathbf{V}_1 \boldsymbol{\Omega} \mathbf{V}_1^T)$ (*vid.* [2.3.33]) y $\mathbf{H}^T \mathbf{H} = \mathbf{P}$ ($mg \times mg$) está definida en [2.3.9]-[2.3.10]. En [3.1.1], $\boldsymbol{\eta}$ ($mn \times 1$) está definido en [2.3.22]-[2.3.23] y $\boldsymbol{\lambda}$ ($mg \times 1$) es la solución del sistema triangular $\mathbf{L}\boldsymbol{\lambda} = (\mathbf{M}^T \tilde{\mathbf{h}})$, donde \mathbf{L} ($mg \times mg$) es el factor de Cholesky de la matriz \mathbf{D} de [3.1.2] y los elementos de $\tilde{\mathbf{h}}$ ($mg \times 1$) están dados en [2.3.19]. El problema de maximizar [3.1.1] equivale a minimizar el opuesto de su logaritmo neperiano (log), que está dado por la siguiente expresión:

$$l(\Phi_i, \Theta_i, \mu, \sigma^2, \mathbf{Q} | \mathbf{w}) = \frac{1}{2} \left[mn \cdot \log(2\pi\sigma^2) + n \cdot \log|\mathbf{Q}| + \log|\mathbf{D}| + \frac{1}{\sigma^2}(\boldsymbol{\eta}^T \boldsymbol{\eta} - \boldsymbol{\lambda}^T \boldsymbol{\lambda}) \right] \quad [3.1.3]$$

En principio, cualquier algoritmo de propósito general puede utilizarse para minimizar [3.1.3]. No obstante, algunas manipulaciones sobre dicha expresión pueden simplificar el problema considerablemente. En concreto, es evidente que el estimador por *MVE* del parámetro σ^2 resulta de igualar a cero la derivada parcial de [3.1.3] respecto a σ^2 . Despejando σ^2 de la expresión resultante:

$$\hat{\sigma}_{MVE}^2 = \frac{\boldsymbol{\eta}^T \boldsymbol{\eta} - \boldsymbol{\lambda}^T \boldsymbol{\lambda}}{mn} \quad [3.1.4]$$

Entonces, sustituyendo σ^2 en [3.1.3] por la expresión [3.1.4], se obtiene el opuesto del logaritmo

neperiano de la *FVE concentrada* [vid. por ejemplo Ansley (1979)], que puede escribirse como:

$$l_*(\Phi_i, \Theta_i, \mu, Q | w) = \frac{mn}{2} \left[\log \left[\frac{2\pi}{mn} \right] + 1 \right] + \frac{n}{2} \log(\Pi_1 \times \Pi_2) \quad [3.1.5]$$

donde:

$$\Pi_1 = (\eta^T \eta - \lambda^T \lambda)^m \quad [3.1.6]$$

$$\Pi_2 = |Q| \times |D|^{\frac{1}{n}} \quad [3.1.7]$$

Por tanto, las estimaciones por *MVE* de los parámetros Φ_i ($i=1,2,\dots,p$), Θ_i ($i=1,2,\dots,q$), μ y Q pueden obtenerse minimizando la función:

$$\Pi(x) = \Pi_1 \times \Pi_2 \quad [3.1.8]$$

donde x es un vector ($k \times 1$) que contiene todos los parámetros a estimar (k). A la vista de [3.1.6] y [3.1.7], la función objetivo [3.1.8] no tiene ninguna estructura especial, por lo que para su minimización numérica debe utilizarse un método de propósito general. El método de *Newton* o algún método *cuasi-Newton* podrían ser los candidatos a tener en cuenta¹.

Alternativamente, la función [3.1.5] puede escribirse como:

$$l_*(\Phi_i, \Theta_i, \mu, Q | w) = \frac{mn}{2} \left[\log \left[\frac{2\pi}{mn} \right] + 1 \right] + \frac{mn}{2} \log(\Delta_1 \times \Delta_2) \quad [3.1.9]$$

donde:

$$\Delta_1 = (\eta^T \eta - \lambda^T \lambda) \quad [3.1.10]$$

$$\Delta_2 = |Q|^{\frac{1}{m}} \times |D|^{\frac{1}{mn}} \quad [3.1.11]$$

Por tanto, las estimaciones por *MVE* de los parámetros Φ_i ($i=1,2,\dots,p$), Θ_i ($i=1,2,\dots,q$), μ y Q también pueden obtenerse minimizando la función:

$$\Delta(x) = \Delta_1 \times \Delta_2 \quad [3.1.12]$$

Entonces, si definimos:

$$\tilde{\eta} = \Delta_2^{1/2} \times \eta \quad [3.1.13]$$

$$\tilde{\lambda} = \Delta_2^{1/2} \times \lambda \quad [3.1.14]$$

es evidente que minimizar [3.1.12] equivale a minimizar la siguiente expresión:

$$\frac{1}{2} \Delta(x) = \frac{1}{2} \tilde{\eta}^T \tilde{\eta} - \frac{1}{2} \tilde{\lambda}^T \tilde{\lambda} = \frac{1}{2} \sum_{i=1}^{mn} \tilde{\eta}_i^2 - \frac{1}{2} \sum_{i=1}^{mg} \tilde{\lambda}_i^2 \quad [3.1.15]$$

Dado que [3.1.15] es la diferencia entre dos sumas de cuadrados, su minimización numérica puede llevarse a cabo utilizando una modificación de algún método específico para sumas de cuadrados (*Gauss-Newton* o *Levenberg-Marquardt*)².

Por tanto, la forma de evaluar la *FVE* descrita en el Capítulo 2, permite, en principio, el empleo para su maximización numérica tanto de algoritmos de propósito general (para minimizar [3.1.8]) como de algoritmos específicos para sumas de cuadrados (para minimizar [3.1.15]). En cualquiera de los dos casos, la estimación de los parámetros Φ_i ($i=1,2,\dots,p$), Θ_i ($i=1,2,\dots,q$), μ y \mathbf{Q} se lleva a cabo *simultáneamente*. Una vez obtenidas dichas estimaciones por cualquiera de los dos procedimientos descritos, la estimación correspondiente de σ^2 se calcula a partir de [3.1.4]. Finalmente, la estimación por *MVE* de la matriz de covarianzas de los residuos del modelo se obtiene multiplicando por [3.1.4] cada componente de la matriz \mathbf{Q} estimada (*vid.* [2.1.11]). A este respecto, es interesante señalar que el procedimiento propuesto por Hillmer y Tiao (1979) no proporciona, estrictamente hablando, un estimador por *MVE* (ni siquiera dando por válida la aproximación a la *FVE* propuesta en ese trabajo), a menos que los elementos de la matriz $\sigma^2 \mathbf{Q}$ sean conocidos (*vid.* Hillmer y Tiao (1979), pp. 659).

3.1.2. Descripción del procedimiento de optimización

El mecanismo de maximización de la *FVE* que se propone en este trabajo, está basado en la minimización de la función objetivo [3.1.8] mediante un método *cuasi-Newton*. El resultado es un algoritmo computacionalmente eficiente, que genera *verdaderas* estimaciones por *MVE* al minimizar la función objetivo respecto a *todos* los parámetros del modelo considerado (incluida la matriz \mathbf{Q}).

El punto de partida para la minimización de [3.1.8] (o de [3.1.15]), consiste en unas estimaciones iniciales de los parámetros Φ_i ($i=1,2,\dots,p$), Θ_i ($i=1,2,\dots,q$), μ y \mathcal{Q} , que se recogen en el vector \mathbf{x}_0 . Correspondientes a dichas estimaciones iniciales, se dispone de dos valores Π_{10} y Π_{20} , de manera que $\Pi(\mathbf{x}_0) = \Pi_{10} \times \Pi_{20}$ (vid. [3.1.8]) [si se estuviera minimizando [3.1.15], se dispondría de $\Delta(\mathbf{x}_0)$]. A continuación, el mecanismo iterativo de minimización de [3.1.8] (o de [3.1.15]), genera una secuencia de puntos \mathbf{x}_{i+1} ($i=0,1,2,\dots$) mediante la siguiente regla:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{d}_i \quad [3.1.16]$$

donde el vector ($k \times 1$) \mathbf{d}_i es una *dirección de búsqueda* y el escalar α_i es una *longitud de paso*, tales que $\Pi(\mathbf{x}_{i+1}) < \Pi(\mathbf{x}_i)$ [o bien $\Delta(\mathbf{x}_{i+1}) < \Delta(\mathbf{x}_i)$]. Si converge, el límite de la secuencia [3.1.16] es aproximadamente un mínimo *local* de la función objetivo (vid. Dennis y Schnabel (1983), cap. 7 para una discusión detallada de distintos criterios de convergencia). La forma de calcular la dirección de búsqueda \mathbf{d}_i en cada iteración es lo que distingue a unos algoritmos de otros. No obstante, en el contexto de algoritmos que utilicen al menos primeras derivadas de la función objetivo, la dirección de búsqueda \mathbf{d}_i suele ser la solución del sistema de k ecuaciones lineales:

$$\mathbf{G}_i \mathbf{d}_i = -\nabla F(\mathbf{x}_i) \quad [3.1.17]$$

donde $\nabla F(\mathbf{x}_i)$ es el vector gradiente de la función objetivo ([3.1.8] ó [3.1.15]) evaluado en \mathbf{x}_i y \mathbf{G}_i es una matriz ($k \times k$) simétrica y no singular. Si, además, \mathbf{G}_i es definida positiva en cada iteración, entonces la dirección \mathbf{d}_i (solución del sistema [3.1.17]) es una *dirección de descenso*³ de la función objetivo en \mathbf{x}_i ($i=0,1,2,\dots$). La importancia de disponer de una dirección de descenso \mathbf{d}_i en cada iteración consiste en que, bajo esta circunstancia, es seguro que existe una longitud de paso $\alpha_i > 0$ tal que el valor de la función objetivo decrece estrictamente al pasar de una iteración a otra. Esto, junto con otras condiciones, garantiza la convergencia de la sucesión [3.1.16] a un mínimo *local* de la función objetivo, independientemente de los valores de las estimaciones iniciales \mathbf{x}_0 (Dennis y Schnabel (1983) se refieren a esta propiedad como *convergencia global*, lo cual *no* debe confundirse con la convergencia de [3.1.16] a un mínimo *global* de la función objetivo).

En nuestro contexto, la utilización de un algoritmo eficiente y globalmente convergente para minimizar [3.1.8] ó [3.1.15], requiere prestar especial atención a ciertos problemas. En primer lugar, para hacer uso de [3.1.17], es necesario evaluar el vector gradiente de la función objetivo en cada

iteración. Aunque pueden obtenerse expresiones analíticas de las primeras derivadas de [3.1.8] y de [3.1.15] (*vid.* Apéndice A.4), la evaluación de dichas expresiones, aunque teóricamente exacta, es muy costosa en términos de tiempo de cálculo. La alternativa consiste en evaluar dichas derivadas mediante *diferencias finitas* de los valores de la función objetivo⁴. La aproximación implícita en esta alternativa supone un importante ahorro computacional y tan sólo una ligera pérdida de precisión numérica (despreciable en todos los casos analizados en este trabajo), aunque para su correcto funcionamiento debe instrumentarse cuidadosamente. En concreto, es muy importante que los valores que puede tomar la función a minimizar estén acotados por un intervalo cuya amplitud sea suficientemente pequeña. Además, esta circunstancia puede mejorar considerablemente el rendimiento global del algoritmo de optimización empleado [*vid.* Shea (1984)].

El segundo problema a tener en cuenta está estrechamente relacionado con el anterior. En la i -ésima iteración del algoritmo empleado para minimizar [3.1.8] (o [3.1.15]), es necesario evaluar la función objetivo [3.1.8] (o los $m(n+g)$ elementos de [3.1.15]) en un conjunto de puntos, generados a partir de x_i , para calcular (i) el vector gradiente de la función objetivo en x_i y (ii) la longitud de paso α_i . En principio, nada garantiza que en todos los puntos de dicho conjunto sea posible evaluar esas cantidades. Concretamente, si alguno de esos puntos corresponde a una matriz \mathbf{Q} no definida positiva o a unos valores numéricos de los parámetros AR que implican no estacionariedad, ni la función objetivo [3.1.8] ni los $m(n+g)$ elementos de [3.1.15] pueden ser evaluados en dicho punto. Una forma sencilla y efectiva de tratar este tipo de situaciones [*vid.* Shea (1984)] consiste en minimizar, en vez de [3.1.8], la siguiente función *normalizada*:

$$F(\mathbf{x}) = \frac{\Pi(\mathbf{x})}{\Pi(\mathbf{x}_0)} = \frac{\Pi_1}{\Pi_{10}} \times \frac{\Pi_2}{\Pi_{20}} \quad [3.1.18]$$

donde Π_{10} y Π_{20} son los valores de [3.1.6] y [3.1.7] en la estimación inicial \mathbf{x}_0 ($\Pi(\mathbf{x}_0) = \Pi_{10} \times \Pi_{20}$). El interés de minimizar [3.1.18] en vez de [3.1.8] consiste en que, si el algoritmo de minimización genera en todas las iteraciones una dirección de descenso, el valor de [3.1.18] está acotado superiormente por 1. En concreto, bajo esta condición, el valor de [3.1.18] en *todos* los puntos generados por el algoritmo será siempre inferior a 1 (también será siempre mayor que cero, ya que los dos factores de [3.1.18] son siempre positivos; *vid.* [3.1.6]-[3.1.7]). Evidentemente, un mínimo de [3.1.18] es también un mínimo de [3.1.8] (una estimación por *MVE* de los parámetros del modelo considerado).

En los casos en los que es necesario evaluar [3.1.18] cuando \mathbf{Q} no es definida positiva o cuando los valores numéricos de los parámetros AR implican no estacionariedad, es suficiente igualar a 1 el valor

de [3.1.18] para (i) poder evaluar numéricamente sus primeras derivadas y (ii) para que el algoritmo de minimización rechace tales puntos como posibles candidatos a ser la solución del problema. Si se desea, esta misma penalización puede imponerse cuando el algoritmo [FVE] detecta que los valores numéricos de los parámetros MA implican no invertibilidad, aunque, salvo en casos de clara ausencia de invertibilidad, no sea estrictamente necesario (*vid.* apartado 2.3.2). Es importante señalar que, en el contexto de la minimización de [3.1.15] mediante un método para sumas de cuadrados, el diseño de una estrategia similar a la descrita no es inmediato, sobre todo a la hora de evaluar numéricamente las derivadas necesarias. Probablemente, sería interesante investigar soluciones en ese contexto.

Para poder aplicar la estrategia descrita es necesario, por tanto, que el algoritmo que se utilice para minimizar [3.1.18] genere en cada iteración una dirección de descenso. Un método *cuasi-Newton* basado en la fórmula BFGS (*vid.* Dennis y Schnabel (1983), cap. 9; Gill, Murray y Wright (1981), cap. 4) es, probablemente, uno de los métodos de optimización numérica más eficientes para generar direcciones de descenso en cada iteración. Aparte de esta característica, un método *cuasi-Newton* basado en la fórmula BFGS tiene otras propiedades que, en el contexto de la estimación de parámetros por MVE, lo hacen especialmente interesante. En concreto, la aproximación en el óptimo a la matriz de segundas derivadas que se actualiza en cada iteración en un método de este tipo, proporciona una forma sencilla y suficientemente precisa para evaluar las covarianzas de las estimaciones obtenidas minimizando [3.1.18]⁵. Los errores estándar estimados que se derivan de esta aproximación (como cualesquiera otros obtenidos por medios teóricamente *exactos*), sólo deberían utilizarse como un indicador del resultado en la inferencia sobre cualquier parámetro estimado. Para contrastar formalmente una hipótesis nula H_0 , consistente en r restricciones sobre los k parámetros considerados, puede utilizarse un test de la *razón de verosimilitudes* (*vid.* por ejemplo Engle (1983); Maddala(1977), pp. 179-80):

$$\xi_{RV} = -2[\log L(\hat{x}_{H_0}) - \log L(\hat{x}_{MVE})] \sim \chi_r^2 \quad [3.1.19]$$

donde $L(\cdot)$ representa el valor de la FVE [3.1.1] evaluada en los argumentos correspondientes. Para evaluar la FVE del proceso ARMA considerado bajo la hipótesis nula H_0 , es necesario estimar los parámetros con las restricciones representadas por H_0 . Siempre que dichas restricciones permitan escribir los parámetros Φ_i ($i=1,2,\dots,p$), Θ_i ($i=1,2,\dots,q$), μ y Q como función *explícita* de los parámetros a estimar bajo H_0 , no es necesario recurrir a métodos numéricos de optimización con restricciones para minimizar [3.1.18] bajo H_0 (*vid.* Gill, Murray y Wright (1983), caps. 5 y 6). En caso contrario, el problema que se plantea puede resultar muy complicado; probablemente, sería interesante investigar posibles soluciones en ese contexto.

Un método de optimización basado en todas estas sugerencias tendría, en definitiva, la siguiente estructura (algoritmo [MAXFV]):

[MAXFV.1] (*Comprobar preestimaciones*). Escoger unas estimaciones iniciales \mathbf{x}_0 y evaluar Π_{10} y Π_{20} según [3.1.6] y [3.1.7], respectivamente, haciendo uso del algoritmo [FVE]. Si las estimaciones iniciales *no* son admisibles, parar; en caso contrario, pasar a [MAXFV.2].

[MAXFV.2] (*Asignar valores iniciales*). Hacer $F(\mathbf{x}_0)=1$; evaluar $\nabla F(\mathbf{x}_0)$; hacer $\mathbf{G}_0=\mathbf{I}_k$ (nótese que el factor de Cholesky de \mathbf{G}_0 es \mathbf{I}_k); hacer $i=0$ y pasar a [MAXFV.3].

[MAXFV.3] (*Paso principal: iteración cuasi-Newton*).

[MAXFV.3.1] (*Dirección de búsqueda*). Resolver el sistema $\mathbf{G}_i\mathbf{d}_i=-\nabla F(\mathbf{x}_i)$.

[MAXFV.3.2] (*Longitud de paso*). Encontrar $\alpha_i>0$ tal que $F(\mathbf{x}_i+\alpha_i\mathbf{d}_i)<F(\mathbf{x}_i)$.

[MAXFV.3.3] (*Nueva estimación*). Hacer $\mathbf{x}_{i+1}=\mathbf{x}_i+\alpha_i\mathbf{d}_i$; evaluar $F(\mathbf{x}_{i+1})$ y $\nabla F(\mathbf{x}_{i+1})$.

[MAXFV.3.4] (*Actualizar \mathbf{G}_i*). Obtener el factor de Cholesky de \mathbf{G}_{i+1} actualizando el factor de Cholesky de \mathbf{G}_i mediante la fórmula BFGS.

[MAXFV.3.5] (*Comprobar convergencia*). Si se cumplen las condiciones de convergencia, pasar a [MAXFV.4]; en caso contrario, hacer $i=i+1$ y volver a [MAXFV.3.1].

[MAXFV.4] (*Estimación de errores estándar, correlaciones y residuos*). En base a las estimaciones obtenidas ($\hat{\mathbf{x}}_{MVE}=\mathbf{x}_{i+1}$), evaluar su matriz de covarianzas y, mediante una última llamada al algoritmo [FVE], evaluar [3.1.4], el vector de residuos y su matriz de covarianzas estimadas.

En el Apéndice A.3 se describe detalladamente el algoritmo [MAXFV] para minimizar [3.1.18], haciendo uso del algoritmo [FVE] descrito en el Capítulo 2 (*vid.* también Apéndice A.2) e incluyendo la posibilidad de incorporar restricciones sencillas del tipo mencionado anteriormente. En resumen, la estructura del algoritmo [MAXFV] consiste en una extensión de un método *cuasi-Newton* (iteración [MAXFV.3]) basado en la fórmula BFGS (paso [MAXFV.3.4]), que contempla las características propias del contexto en el que se está aplicando. Básicamente, estas características consisten en (i) la incorporación de la estrategia descrita anteriormente para evaluar la función objetivo en puntos no admisibles, detectados por el algoritmo [FVE] (pasos [MAXFV.3.2] y [MAXFV.3.3]), (ii) la estimación, una vez el algoritmo ha convergido, de los errores estándar y las correlaciones entre los parámetros (paso [MAXFV.4]) y (iii) la evaluación de los residuos correspondientes y de su matriz de covarianzas (paso [MAXFV.4]) (los detalles pueden encontrarse en el Apéndice A.3).

3.1.3. Análisis y aplicaciones del procedimiento de estimación

Para contrastar empíricamente la validez de los métodos propuestos (algoritmos **[FVE]** y **[MAXFV]**) para la estimación por *MVE* de procesos ARMA multivariantes, se han estimado, en primer lugar, una serie de modelos diagonales (por sencillez y claridad), seleccionados de la Tabla 2.4.1 (*vid.* Sección 2.4), simulados según se describe en el Apéndice A.5. Todos los modelos estimados tienen la forma:

$$\Phi(B)w_t = \Theta(B)a_t \quad [3.1.20]$$

con:

$$\Phi(B) = \begin{bmatrix} 1 - \phi_{11}B - \phi_{12}B^2 & 0 \\ 0 & 1 - \phi_{21}B - \phi_{22}B^2 \end{bmatrix} \begin{bmatrix} 1 - \Phi_1 B^S & 0 \\ 0 & 1 - \Phi_2 B^S \end{bmatrix} \quad [3.1.21]$$

$$\Theta(B) = \begin{bmatrix} 1 - \theta_{11}B - \theta_{12}B^2 & 0 \\ 0 & 1 - \theta_{21}B - \theta_{22}B^2 \end{bmatrix} \begin{bmatrix} 1 - \Theta_1 B^S & 0 \\ 0 & 1 - \Theta_2 B^S \end{bmatrix} \quad [3.1.22]$$

$$E[a_t a_t^T] = \begin{bmatrix} \sigma_{11}^2 & \sigma_{12}^2 \\ \sigma_{12}^2 & \sigma_{22}^2 \end{bmatrix} \quad [3.1.23]$$

donde, dependiendo del modelo considerado, algunos parámetros de [3.1.21] y [3.1.22] son iguales a cero. Cada modelo (12 en total) se ha estimado con 100 realizaciones independientes de tamaño $n=50$ (excepto $n=100$ para los modelos con período estacional $S=12$) utilizando los algoritmos **[MAXFV]** y **[FVE]** (1200 estimaciones en total). Los resultados de cada simulación se encuentran en las Tablas 3.1.1-3.1.12, que contienen, para cada modelo, los valores teóricos de los parámetros, las medias y los errores estándar muestrales de la estimaciones obtenidas por *MVE*.

La brevedad en el tamaño de las muestras consideradas y los valores numéricos de los parámetros (muchos de ellos cerca de las fronteras de no invertibilidad o no estacionariedad), obedecen a dos propósitos: (i) contrastar la validez de los algoritmos propuestos en situaciones *difíciles* y (ii) comparar las propiedades de distintos estimadores por máxima verosimilitud en dichas situaciones (*vid.* sección 3.3). Es lógico esperar que si los algoritmos propuestos funcionan bien en estos casos, también lo harán en situaciones mejor condicionadas [en concreto, para muestras largas no cabe esperar grandes diferencias en las estimaciones obtenidas por criterios distintos al de *MVE*; *vid.* Ansley y Newbold (1980)].

M 1	ϕ_{11}	ϕ_{21}	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	0.95	0.40	1.00	1.00	2.00
<i>Valor medio estimado</i>	0.923 (0.041)	0.383 (0.090)	1.003 (0.194)	0.988 (0.245)	1.959 (0.423)

Tabla 3.1.1. Resultados de la estimación por MVE para un AR(1) bivalente diagonal ($n=50$; 100 repeticiones).

M 2	ϕ_{11}	ϕ_{12}	ϕ_{21}	ϕ_{22}	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	0.95	-0.96	0.85	-0.95	1.00	1.00	2.00
<i>Valor medio estimado</i>	0.941 (0.045)	-0.939 (0.038)	0.839 (0.037)	-0.928 (0.042)	0.999 (0.194)	0.995 (0.248)	1.950 (0.424)

Tabla 3.1.2. Resultados de la estimación por MVE para un AR(2) bivalente diagonal ($n=50$; 100 repeticiones).

M 3	θ_{11}	θ_{21}	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	0.30	0.70	1.00	1.00	2.00
<i>Valor medio estimado</i>	0.306 (0.122)	0.703 (0.082)	1.000 (0.194)	0.984 (0.245)	1.947 (0.424)

Tabla 3.1.3. Resultados de la estimación por MVE para un MA(1) bivalente diagonal ($n=50$; 100 repeticiones).

M 4	θ_{11}	θ_{12}	θ_{21}	θ_{22}	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	0.95	-0.96	0.85	-0.95	1.00	1.00	2.00
<i>Valor medio estimado</i>	0.955 (0.066)	-0.970 (0.061)	0.858 (0.072)	-0.961 (0.074)	0.965 (0.189)	0.956 (0.232)	1.896 (0.417)

Tabla 3.1.4. Resultados de la estimación por MVE para un MA(2) bivariante diagonal ($n=50$; 100 repeticiones).

M 5	ϕ_{11}	ϕ_{21}	θ_{11}	θ_{21}	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	0.90	0.30	0.70	0.30	1.00	1.00	2.00
<i>Valor medio estimado</i>	0.820 (0.161)	0.214 (0.507)	0.619 (0.217)	0.211 (0.532)	0.992 (0.195)	0.986 (0.245)	1.941 (0.426)

Tabla 3.1.5. Resultados de la estimación por MVE para un ARMA(1,1) bivariante diagonal ($n=50$; 100 repeticiones).

M 5	ϕ_{11}	ϕ_{21}	θ_{11}	θ_{21}	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	-0.90	-0.80	0.40	0.50	1.00	1.00	2.00
<i>Valor medio estimado</i>	-0.875 (0.066)	-0.770 (0.073)	0.414 (0.124)	0.516 (0.102)	0.994 (0.195)	0.987 (0.244)	1.943 (0.424)

Tabla 3.1.6. Resultados de la estimación por MVE para un ARMA(1,1) bivariante diagonal ($n=50$; 100 repeticiones).

M 15	ϕ_{11}	ϕ_{21}	Φ_1	Φ_2	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	0.20	0.40	0.90	0.40	1.00	1.00	2.00
<i>Valor medio estimado</i>	0.199	0.399	0.882	0.387	0.990	0.978	1.934
<i>Error estándar estimado</i>	(0.113)	(0.107)	(0.050)	(0.100)	(0.194)	(0.253)	(0.428)

Tabla 3.1.7. Resultados de la estimación por MVE para un $AR(1) \times AR(1)_4$ bivalente diagonal ($n=50$; 100 repeticiones).

M 18	ϕ_{11}	ϕ_{21}	Φ_1	Φ_2	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	0.90	0.40	0.20	0.95	1.00	1.00	2.00
<i>Valor medio estimado</i>	0.881	0.384	0.187	0.945	0.991	0.991	1.967
<i>Error estándar estimado</i>	(0.040)	(0.068)	(0.083)	(0.015)	(0.126)	(0.154)	(0.277)

Tabla 3.1.8. Resultados de la estimación por MVE para un $AR(1) \times AR(1)_{12}$ bivalente diagonal ($n=100$; 100 repeticiones).

M 22	θ_{11}	θ_{21}	Θ_1	Θ_2	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	0.60	0.80	0.70	0.90	1.00	1.00	2.00
<i>Valor medio estimado</i>	0.603	0.798	0.712	0.927	0.973	0.969	1.879
<i>Error estándar estimado</i>	(0.110)	(0.083)	(0.118)	(0.104)	(0.191)	(0.245)	(0.412)

Tabla 3.1.9. Resultados de la estimación por MVE para un $MA(1) \times MA(1)_4$ bivalente diagonal ($n=50$; 100 repeticiones).

M 25	θ_{11}	θ_{21}	θ_1	θ_2	σ_{11}^2	σ_{12}^2	σ_{22}^2
Valor del parámetro	0.90	0.70	0.90	0.80	1.00	1.00	2.00
Valor medio estimado	0.916 (0.047)	0.709 (0.058)	0.904 (0.112)	0.819 (0.115)	0.980 (0.139)	0.968 (0.158)	1.906 (0.301)

Tabla 3.1.10. Resultados de la estimación por MVE para un $MA(1) \times MA(1)_{12}$ bivariente diagonal ($n=100$; 100 repeticiones).

M 11	ϕ_1	ϕ_2	θ_1	θ_2	σ_{11}^2	σ_{12}^2	σ_{22}^2
Valor del parámetro	-0.90	-0.80	0.40	0.50	1.00	1.00	2.00
Valor medio estimado	-0.877 (0.056)	-0.771 (0.080)	0.408 (0.153)	0.522 (0.133)	0.998 (0.203)	0.987 (0.248)	1.910 (0.409)

Tabla 3.1.11. Resultados de la estimación por MVE para un $ARMA(1,1)_4$ bivariente diagonal ($n=50$; 100 repeticiones).

M 14	ϕ_1	ϕ_2	θ_1	θ_2	σ_{11}^2	σ_{12}^2	σ_{22}^2
Valor del parámetro	-0.90	-0.80	0.40	0.50	1.00	1.00	2.00
Valor medio estimado	-0.899 (0.030)	-0.799 (0.045)	0.397 (0.100)	0.502 (0.101)	1.008 (0.122)	1.002 (0.149)	1.979 (0.261)

Tabla 3.1.12. Resultados de la estimación por MVE para un $ARMA(1,1)_{12}$ bivariente diagonal ($n=100$; 100 repeticiones).

En todos los casos considerados, las diferencias entre los valores teóricos de los parámetros y los valores medios estimados son muy pequeñas. Por ello, los valores teóricos de los parámetros se encuentran siempre en un intervalo, alrededor de sus valores medios estimados, de amplitud igual a tan sólo dos errores estándar. Esta circunstancia se refleja también en el error relativo medio cometido en la estimación de cada modelo, calculado como el cociente entre la suma de los errores relativos asociados a cada parámetro y el número total de parámetros estimados en cada modelo. En ningún caso, el error relativo medio es superior al 2.8%, excepto en el modelo sobreparametrizado de la Tabla 3.1.5; en este caso, el error relativo medio alcanza el 12.0%, lo cual está de acuerdo con los grandes errores estándar asociados a los parámetros redundantes.

Por supuesto, estos resultados no se restringen al caso de modelos diagonales. A modo de ilustración, se han estimado dos modelos bivariantes, cada uno de ellos con 100 realizaciones independientes de tamaño $n=100$, generadas según se describe en el Apéndice A.5. Los modelos estimados han sido un MA(2):

$$w_t = (I_2 - \Theta_1 B - \Theta_2 B^2) a_t \quad [3.1.24]$$

con:

$$\Theta_1 = \begin{bmatrix} -1.00 & -1.80 \\ 0.50 & 1.00 \end{bmatrix}; \Theta_2 = \begin{bmatrix} -0.80 & -1.40 \\ 0.40 & 0.70 \end{bmatrix}; E[a_t a_t^T] = \begin{bmatrix} 1.00 & \\ & 1.00 \ 2.00 \end{bmatrix} \quad [3.1.25]$$

y un ARMA (1,1):

$$(I_2 - \Phi_1 B) w_t = (I_2 - \Theta_1 B) a_t \quad [3.1.26]$$

con:

$$\Phi_1 = \begin{bmatrix} 1.10 & 0.50 \\ -1.00 & 0.40 \end{bmatrix}; \Theta_1 = \begin{bmatrix} 1.00 & 1.00 \\ -0.80 & -1.20 \end{bmatrix}; E[a_t a_t^T] = \begin{bmatrix} 1.00 & \\ & 1.00 \ 2.00 \end{bmatrix} \quad [3.1.27]$$

Los resultados (medias y errores estándar muestrales) de las 100 estimaciones por MVE para el proceso MA(2) [3.1.24]-[3.1.25] son los siguientes:

$$\hat{\Theta}_1 = \begin{bmatrix} -1.00444 & -1.80205 \\ (0.09690) & (0.06039) \\ 0.49525 & 1.00033 \\ (0.08539) & (0.05867) \end{bmatrix} \quad [3.1.28]$$

$$\hat{\theta}_2 = \begin{bmatrix} -0.81131 & -1.40317 \\ (0.17570) & (0.11880) \\ 0.40894 & 0.70187 \\ (0.14342) & (0.09402) \end{bmatrix} \quad [3.1.29]$$

$$E[\hat{a}_t \hat{a}_t^T] = \begin{bmatrix} 0.97809 & \\ (0.12805) & \\ 0.99949 & 1.96114 \\ (0.16319) & (0.29378) \end{bmatrix} \quad [3.1.30]$$

Y para el proceso ARMA(1,1) [3.1.26]-[3.1.27]:

$$\hat{\phi}_1 = \begin{bmatrix} 1.09940 & 0.51538 \\ (0.05432) & (0.04248) \\ -0.97760 & 0.38480 \\ (0.10193) & (0.07739) \end{bmatrix} \quad [3.1.31]$$

$$\hat{\theta}_1 = \begin{bmatrix} 1.00543 & 1.02248 \\ (0.08782) & (0.08462) \\ -0.77610 & -1.20312 \\ (0.11808) & (0.07670) \end{bmatrix} \quad [3.1.32]$$

$$E[\hat{a}_t \hat{a}_t^T] = \begin{bmatrix} 0.98058 & \\ (0.13201) & \\ 0.98418 & 1.93134 \\ (0.16752) & (0.29255) \end{bmatrix} \quad [3.1.33]$$

En conjunto, la precisión de las estimaciones obtenidas es considerable. Como antes, los valores teóricos de todos los parámetros se encuentran siempre en un intervalo, alrededor de sus valores medios estimados, de amplitud igual a sólo dos errores estándar. Para concluir, cabe señalar que el error relativo medio en la estimación del proceso ARMA(1,1) es de un 2.0% y tan sólo de un 0.8% en la del MA(2).

Los resultados anteriores sugieren con claridad que los algoritmos [FVE] y [MAXFV] resultan adecuados para sus fines. No obstante, para ilustrar el uso en la práctica de las ventajas que presentan estos algoritmos frente a otros alternativos, se han aplicado dichos procedimientos a una situación con datos reales (que también se emplea en las secciones siguientes). En Flores (1987) y Cancio (1989) se estima un modelo cuatrivariante de la macroeconomía de los Estados Unidos, con la siguiente estructura:

$$\begin{pmatrix} 1-\phi_{11}B & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1-\phi_{13}B & 0 \\ 0 & 0 & 0 & (1-\phi_{14}B-\phi_{24}B^2)(1-\alpha B) \end{pmatrix} \begin{pmatrix} sr_t \\ pib_t - \mu \\ r_t \\ m1_t \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\theta_1B & 0 \\ 0 & 1 & -\theta_2B & -\theta_3B \\ 0 & 0 & 1 & -\theta_4B \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_{1t} \\ a_{2t} \\ a_{3t} \\ a_{4t} \end{pmatrix} \quad [3.1.34]$$

donde $sr_t = \nabla \log(SR_t)$, $pib_t = \nabla \log(PIB_t)$, $r_t = \nabla \log(1 + R_t)$ y $m1_t = \nabla \log(M1_t)$ (vid. Flores (1987) para la definición y fuente de las variables SR_t , PIB_t , R_t y $M1_t$). Las principales características del modelo [3.1.34] pueden resumirse en (i) la presencia del factor $(1 - \alpha B)$ en la posición (4,4) de la matriz AR, cuya estacionariedad se desea contrastar (vid. Flores (1987), pp. 177), (ii) el elevado número de parámetros a estimar (20, incluyendo la matriz de covarianzas de los residuos) y (iii) la brevedad en el tamaño de la muestra considerada ($n=31$ observaciones anuales: 1953-1983). Si $\alpha=1$, la representación [3.1.34] se simplifica eliminando el factor $(1 - \alpha B)$ y reemplazando $m1_t = \nabla \log(M1_t)$ por $m1_t = \nabla^2 \log(M1_t)$ [lo cual implica *neutralidad monetaria*; vid. Flores (1987)].

En la Tabla 3.1.13 se presentan las estimaciones, obtenidas con el algoritmo [MAXFV], del modelo [3.1.34] en dos situaciones: (i) imponiendo $\alpha=1$ [eliminando de [3.1.34] el factor $(1 - \alpha B)$ y reemplazando $m1_t = \nabla \log(M1_t)$ por $m1_t = \nabla^2 \log(M1_t)$] y (ii) no imponiendo dicha restricción [estimando *todos* los parámetros de [3.1.34] con $m1_t = \nabla \log(M1_t)$]. En esta tabla también se presenta el valor del logaritmo neperiano de la *FVE* evaluada en ambos puntos del espacio paramétrico.

	μ	ϕ_{11}	ϕ_{13}	ϕ_{14}	ϕ_{24}	α	θ_1	θ_2	θ_3	θ_4	<i>log FVE</i>
Estimaciones con $\alpha=1$	0.029 (0.003)	0.236 (0.088)	0.362 (0.083)	-0.536 (0.069)	-0.339 (0.065)	* *	0.503 (0.351)	1.782 (0.515)	-0.746 (0.141)	-0.522 (0.045)	368.545
Estimaciones con α libre	0.029 (0.003)	0.244 (0.083)	0.353 (0.080)	-0.542 (0.066)	-0.354 (0.064)	0.997 (0.004)	0.546 (0.302)	1.976 (0.513)	-0.714 (0.144)	-0.492 (0.043)	375.964

Tabla 3.1.13. Estimaciones por *MVE* del modelo [3.1.34] con y sin la restricción $\alpha=1$, obtenidas con el algoritmo [MAXFV] (un * indica que el parámetro correspondiente no se ha estimado).

La estimación de la matriz de covarianzas de los residuos, correspondiente al modelo estimado con $\alpha=1$, es la siguiente:

$$E[\hat{\mathbf{a}}_t \hat{\mathbf{a}}_t^T] = \begin{pmatrix} 0.56 \times 10^{-3} & & & & \\ 0.80 \times 10^{-4} & 0.16 \times 10^{-3} & & & \\ -0.75 \times 10^{-4} & -0.16 \times 10^{-4} & 0.33 \times 10^{-4} & & \\ 0.41 \times 10^{-3} & 0.72 \times 10^{-4} & -0.43 \times 10^{-4} & 0.35 \times 10^{-3} & \end{pmatrix} \quad [3.1.35]$$

mientras que la correspondiente a la estimación que no impone $\alpha=1$ es⁶:

$$E[\hat{\mathbf{a}}_t \hat{\mathbf{a}}_t^T] = \begin{pmatrix} 0.55 \times 10^{-3} & & & & \\ 0.57 \times 10^{-4} & 0.18 \times 10^{-3} & & & \\ -0.75 \times 10^{-4} & -0.54 \times 10^{-5} & 0.34 \times 10^{-4} & & \\ 0.41 \times 10^{-3} & 0.58 \times 10^{-4} & -0.43 \times 10^{-4} & 0.36 \times 10^{-3} & \end{pmatrix} \quad [3.1.36]$$

En conjunto, los resultados de ambas estimaciones *no* pueden considerarse significativamente distintos. Esta conclusión puede justificarse, además, examinando en primer lugar las Figuras 3.1.1-3.1.8, que contienen los residuos (estandarizados) correspondientes a ambas estimaciones (evaluados según el algoritmo [CRES] del apartado 2.3.1), sus medias y sus desviaciones típicas *muestrales*. Este examen no sugiere diferencias significativas entre los residuos de las dos estimaciones y, además, justifica la inclusión del parámetro μ en [3.1.34], ya que la no inclusión de dicho parámetro podría generar unos residuos \hat{a}_{2t} con media no nula [como ocurre con los resultados obtenidos por Cancio (1989)].

Otro argumento favorable a la conclusión citada anteriormente, puede obtenerse examinando los perfiles de la *FVE* alrededor de la estimación puntual de cada uno de los parámetros de la especificación [3.1.34]. Este examen proporciona, además, una comprobación de la convergencia del algoritmo [MAXFV] a un máximo local de la *FVE*. Con este propósito, en las Figuras 3.1.9-3.1.18 se representa, para cada parámetro (x_i), el cociente entre (i) el valor del logaritmo neperiano de la *FVE*, evaluada en 200 puntos que recorren un rango de ± 4.0 errores estándar alrededor de cada estimación puntual [$\log L(x_i | \mathbf{x}^*)$] y (ii) el valor del logaritmo neperiano de la *FVE* evaluada en la estimación por *MVE* [$\log L(\mathbf{x}^*)$]. En todos los casos, el rango de variación de este cociente se ha representado entre los mismos límites (0.975 y 1.0). Por último, en cada gráfico también se representa el intervalo de confianza del 95.0% para cada parámetro (aunque, como se discute más abajo, no tiene sentido calcular de la forma habitual un intervalo de confianza para el parámetro α si su verdadero valor es la unidad).

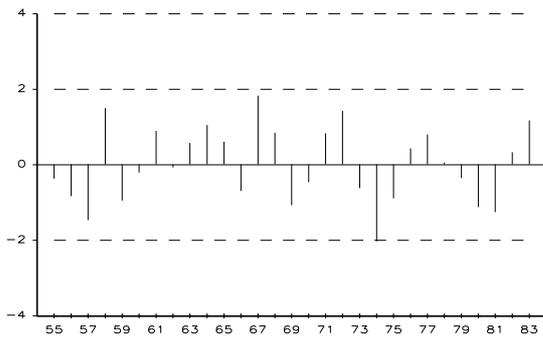


Fig. 3.1.1: \hat{a}_{1t} (con $\alpha=1$); $\bar{a}_{1t}=0.001$ (0.004); $\hat{\sigma}_{1t}=0.024$.

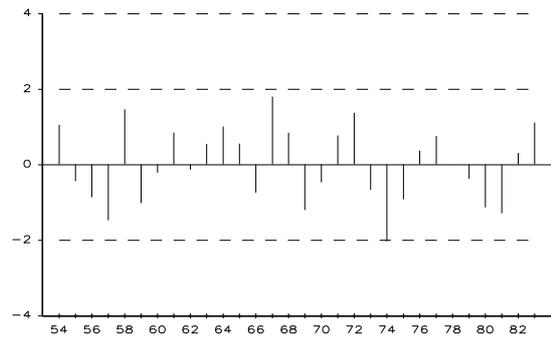


Fig. 3.1.5: \hat{a}_{1t} (con α libre); $\bar{a}_{1t}=0.002$ (0.004); $\hat{\sigma}_{1t}=0.024$.

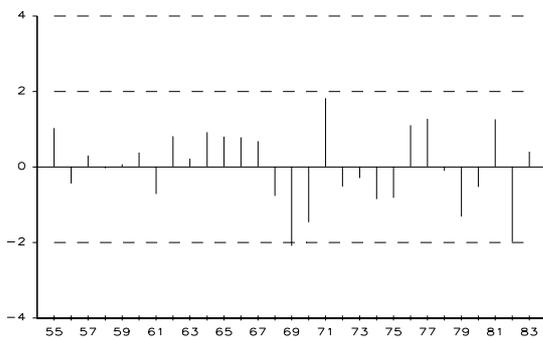


Fig. 3.1.2: \hat{a}_{2t} (con $\alpha=1$); $\bar{a}_{2t}=0.001$ (0.002); $\hat{\sigma}_{2t}=0.013$.

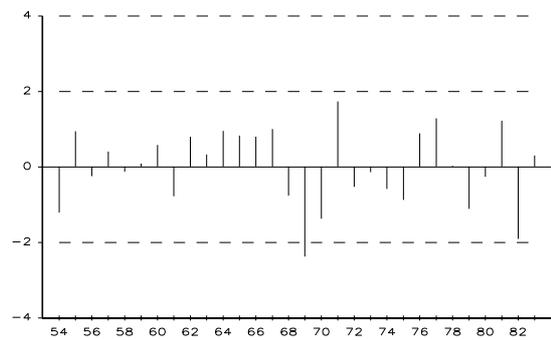


Fig. 3.1.6: \hat{a}_{2t} (con α libre); $\bar{a}_{2t}=0.001$ (0.002); $\hat{\sigma}_{2t}=0.013$.

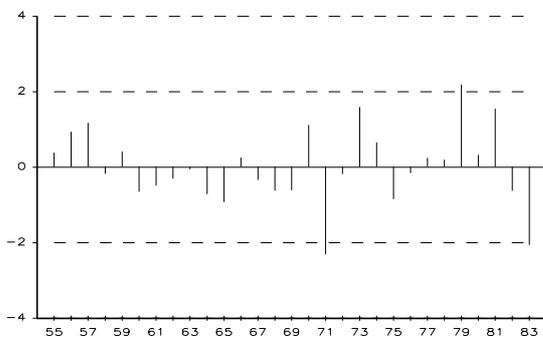


Fig. 3.1.3: \hat{a}_{3t} (con $\alpha=1$); $\bar{a}_{3t}=0.000$ (0.001); $\hat{\sigma}_{3t}=0.006$.

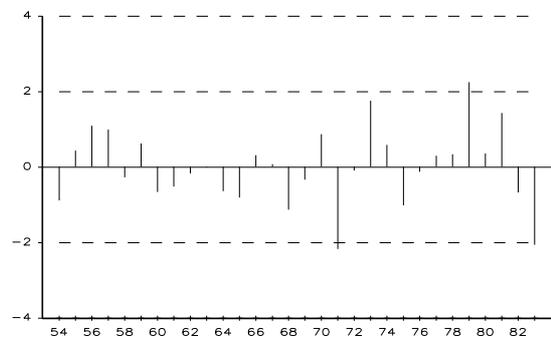


Fig. 3.1.7: \hat{a}_{3t} (con α libre); $\bar{a}_{3t}=0.000$ (0.001); $\hat{\sigma}_{3t}=0.006$.

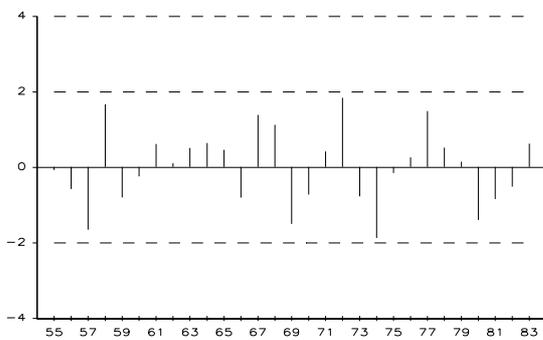


Fig. 3.1.4: \hat{a}_{4t} (con $\alpha=1$); $\bar{a}_{4t}=0.003$ (0.003); $\hat{\sigma}_{4t}=0.019$.

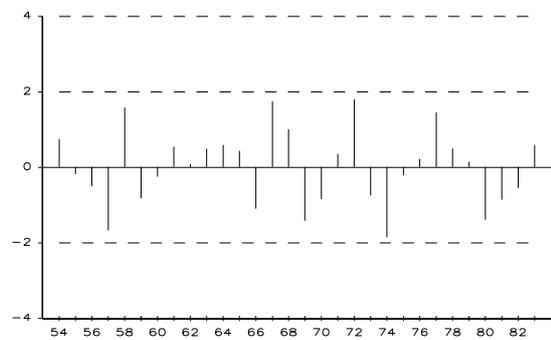


Fig. 3.1.8: \hat{a}_{4t} (con α libre); $\bar{a}_{4t}=0.003$ (0.003); $\hat{\sigma}_{4t}=0.019$.

Estas representaciones pueden resultar útiles para contrastar, además de la validez de las estimaciones puntuales, la validez de los errores estándar calculados según el procedimiento descrito en el apartado 3.1.2. Los rangos de variación de las cantidades representadas en cada eje implican que, si los errores estándar de los parámetros están correctamente estimados, todos los perfiles deben presentar una apariencia similar: la de una campana (por normalidad) centrada en la estimación puntual, cuya base ocupa la práctica totalidad del eje horizontal (suponiendo que la *FVE* pueda evaluarse en todos los puntos representados en dicho eje; *vid.* Fig. 3.1.14). Un perfil más apuntado (más estrecho) podría indicar que el error estándar del parámetro correspondiente ha sido sobreestimado, mientras que un perfil menos apuntado (más ancho) podría sugerir la subestimación de dicho error estándar.

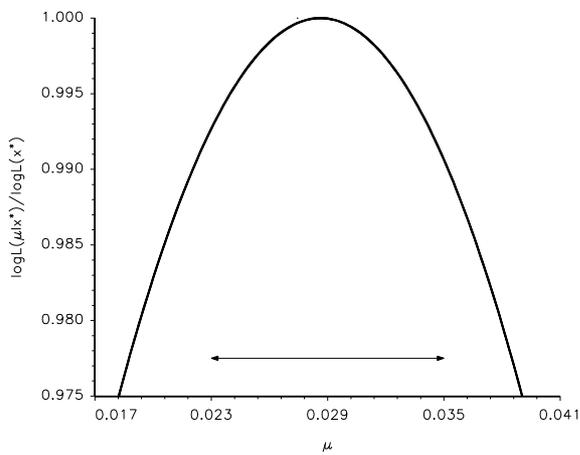


Fig. 3.1.9: Perfil para μ [0.029 (0.003)].

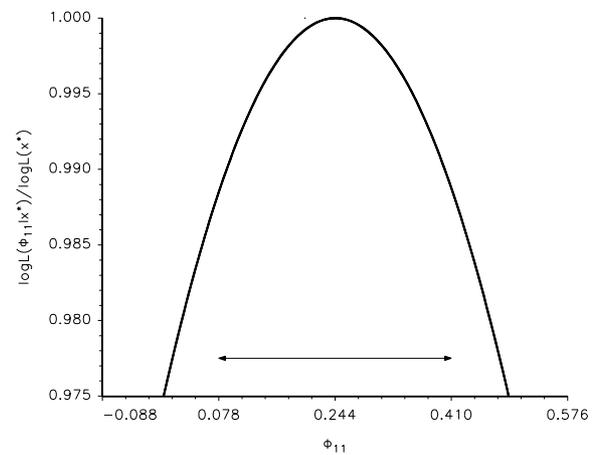


Fig. 3.1.10: Perfil para ϕ_{11} [0.244 (0.083)].

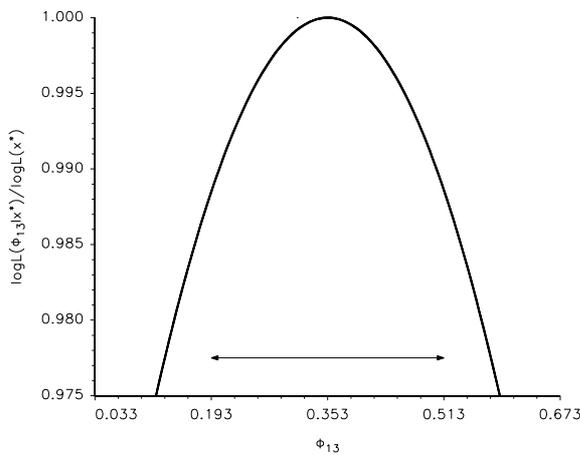


Fig. 3.1.11: Perfil para ϕ_{13} [0.353 (0.080)].

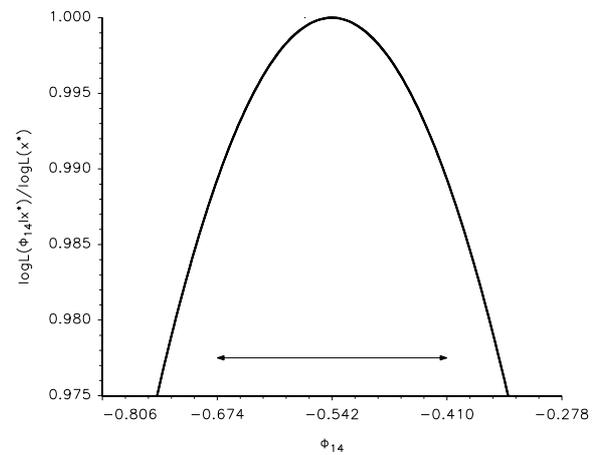


Fig. 3.1.12: Perfil para ϕ_{14} [-0.542 (0.066)].

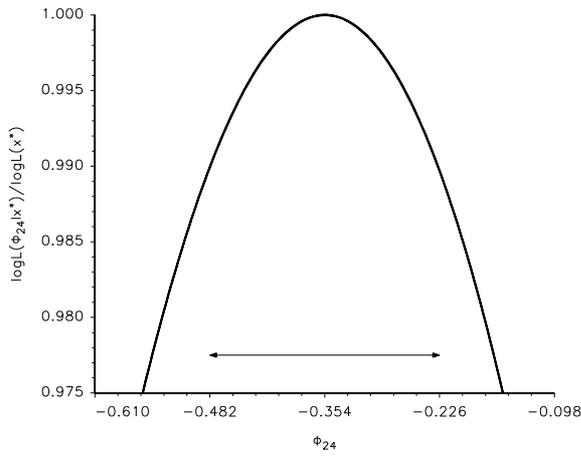


Fig. 3.1.13: Perfil para ϕ_{24} $[-0.354 (0.064)]$.

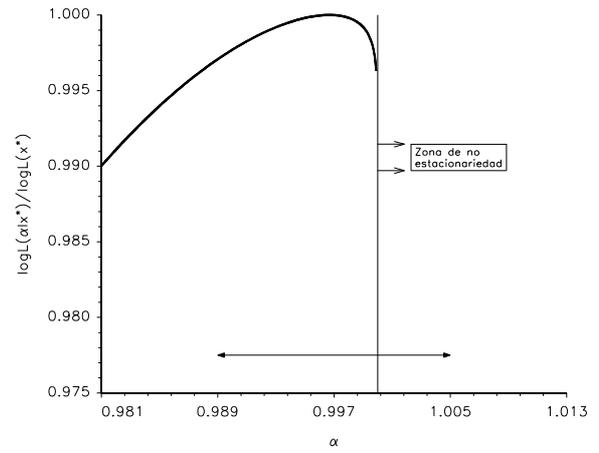


Fig. 3.1.14: Perfil para α $[0.997 (0.004)]$.

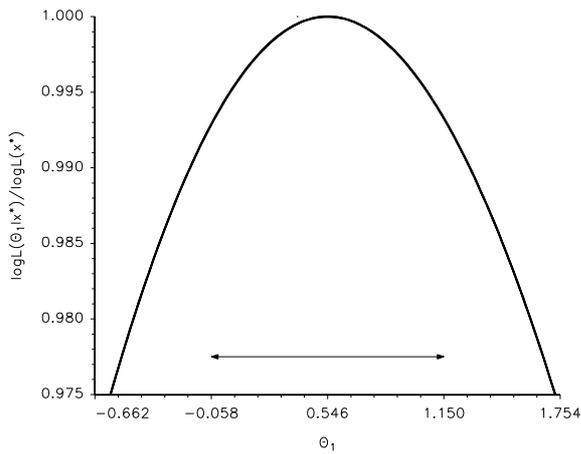


Fig. 3.1.15: Perfil para θ_1 $[0.546 (0.302)]$.

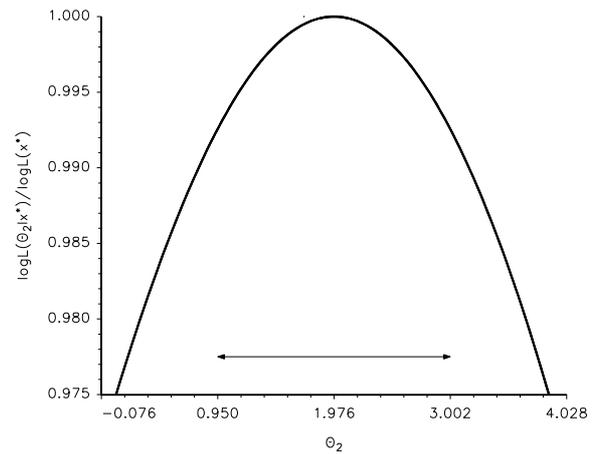


Fig. 3.1.16: Perfil para θ_2 $[1.976 (0.513)]$.

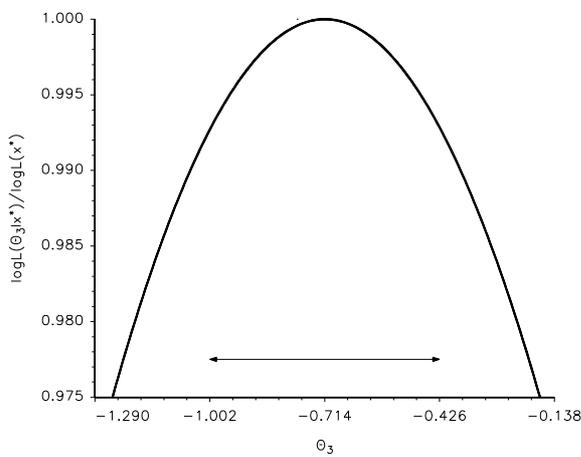


Fig. 3.1.17: Perfil para θ_3 $[-0.714 (0.144)]$.

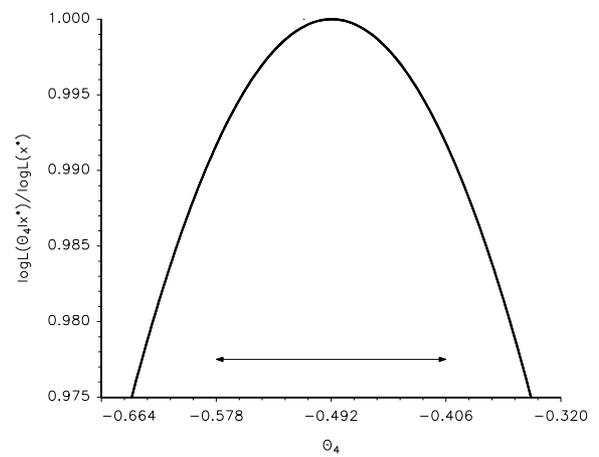


Fig. 3.1.18: Perfil para θ_4 $[-0.492 (0.043)]$.

Las Figuras 3.1.10-3.1.13 podrían indicar una ligera sobreestimación de los errores estándar correspondientes a los parámetros autorregresivos (que, en cualquier caso, resultan claramente significativos). Por su parte, la Figura 3.1.14 sugiere una clara subestimación del error estándar correspondiente al parámetro α . Esto es razonable cuando el verdadero valor de dicho parámetro es 1, ya que, en tal caso, la *FVE* no satisface las condiciones de regularidad necesarias para justificar las propiedades asintóticas del estimador por *MVE* (vid. por ejemplo Peña (1989), pp. 611-12). La Figura 3.1.14 ilustra claramente esta última observación (nótese que la *FVE* no está definida para $\alpha \geq 1$). Todas estas consideraciones sugieren concluir que el parámetro α *no* es significativamente distinto de la unidad, por lo que la especificación [3.1.34] sin el factor $(1 - \alpha B)$ y con $m1_t = \nabla^2 \log(M1_t)$ puede considerarse adecuada [esta conclusión es la misma que la obtenida por Flores (1987), aunque no por Cancio (1989)].

3.1.4. Notas para la sección 3.1

[1] Para minimizar numéricamente un campo escalar $F: E^n \rightarrow E$ dos veces diferenciable, suelen emplearse algoritmos de optimización no lineal con la siguiente estructura:

PASO INICIAL: escoger una estimación inicial x_0 de la solución y unos criterios de convergencia (tolerancias, máximo número de iteraciones, ...); hacer $k=0$ y ejecutar el paso principal.

PASO PRINCIPAL:

[P.1] evaluar $F(x_k)$, $\nabla F(x_k)$ y G_k , donde $\nabla F(x_k)$ es el gradiente de $F(x_k)$ y G_k es una matriz simétrica y no singular;

[P.2] calcular una *dirección de búsqueda* d_k resolviendo el sistema lineal $G_k d_k = -\nabla F(x_k)$;

[P.3] calcular una *longitud de paso* α_k tal que $F(x_k + \alpha_k d_k) < F(x_k)$ (vid. nota [3] más abajo);

[P.5] obtener una nueva estimación de la solución óptima como $x_{k+1} = x_k + \alpha_k d_k$;

[P.6] si se cumplen los criterios de convergencia, parar; en caso contrario, hacer $k = k+1$ y volver a [P.1].

En este contexto, la diferencia entre unos algoritmos y otros reside en el cálculo de la matriz G_k de [P.1] y [P.2] que, a su vez, determina la dirección de búsqueda en cada iteración. Si $G_k = I_n$, el método resultante es el del gradiente; este método (i) sólo requiere primeras derivadas, (ii) genera en cada iteración una *dirección de descenso* (vid. nota [3]) y (iii) presenta una tasa de convergencia *global lineal*. Por otro lado, si $G_k = H(x_k)$ [matriz hessiana de $F(x_k)$], tenemos el método de Newton, que (i) requiere primeras y segundas derivadas, (ii) en general, debe ser modificado para generar direcciones de descenso (ya que $H(x_k)$ no siempre es definida positiva) y (iii) presenta una tasa de convergencia *local cuadrática* (*global* con ciertas modificaciones). Por último, cuando G_k es una *aproximación definida positiva* (vid. nota [5]) a $H(x_k)$, tenemos un método cuasi-Newton, que (i) sólo requiere primeras derivadas, (ii) genera siempre direcciones de descenso y (iii) converge *global* y *superlinealmente* (vid. Gill, Murray y Wright (1981), pp. 56-8 y 99-133; Dennis y Schnabel (1983), pp. 5 y caps. 6, 7 y 9).

[2] Para minimizar numéricamente un campo escalar $F:E^n \rightarrow E$ dos veces diferenciable, definido como la suma de m campos escalares al cuadrado [$F(x) = \frac{1}{2} \sum_{i=1}^m f_i(x)^2$], resulta conveniente sacar provecho de la estructura especial que en este caso tienen el vector gradiente $\nabla F(x_k)$ y la matriz hessiana $H(x_k)$ de la función objetivo. Resulta inmediato comprobar que $\nabla F(x_k) = J(x_k)^T f(x_k)$ y que $H(x_k) = J(x_k)^T J(x_k) + M(x_k)$, donde $f(x_k) = [f_1(x_k), f_2(x_k), \dots, f_m(x_k)]^T$, $J(x_k)_{ij} = \partial f_i(x_k) / \partial x_j$ ($i=1,2,\dots,m; j=1,2,\dots,n$) y $M(x_k) = \sum_{i=1}^m f_i(x_k) H_i(x_k)$, siendo $H_i(x_k)$ la matriz hessiana de $f_i(x_k)$. Cuando en el algoritmo tipo de la nota [1] $G_k = J(x_k)^T J(x_k)$, se obtiene el método de Gauss-Newton para sumas de cuadrados, cuyas propiedades de convergencia son similares a las del método de Newton cuando el término de segundo orden $M(x_k)$ es suficientemente pequeño (a pesar de que sólo se utilizan primeras derivadas). Por otro lado, si $G_k = (J(x_k)^T J(x_k) + \lambda_k I_n)$, el método resultante es el de Levenberg-Marquardt, que, manipulando adecuadamente el valor del parámetro $\lambda_k > 0$, implica un compromiso entre el método de Gauss-Newton (cuando $\lambda_k \rightarrow 0$) y el del gradiente (cuando $\lambda_k \rightarrow \infty$) (vid. Gill, Murray y Wright (1981), pp. 133-8; Dennis y Schnabel (1983), cap. 10).

[3] Si un campo escalar $F:E^n \rightarrow E$ es diferenciable en $x_k \in E^n$, entonces d_k es una *dirección de descenso* de F en x_k sí y sólo sí existe un escalar $\delta_k > 0$ tal que $F(x_k + \alpha_k d_k) < F(x_k)$ para todo $\alpha_k \in (0, \delta_k)$. A partir de esta definición, resulta inmediato comprobar que si $\nabla F(x_k)^T d_k < 0$, entonces d_k es una dirección de descenso de F en x_k . Además, en el contexto del algoritmo tipo de la nota [1], donde $d_k = -G_k^{-1} \nabla F(x_k)$, resulta evidente que si G_k es definida positiva en *todas* las iteraciones, entonces $\nabla F(x_k)^T d_k < 0$, por lo que las direcciones de búsqueda d_k generadas por el algoritmo son siempre direcciones de descenso. Por este motivo resulta fundamental garantizar el carácter definido positivo de la matriz G_k en cada iteración, lo cual requiere especial atención en los métodos de Newton, Gauss-Newton y Levenberg-Marquardt (vid. por ejemplo Gill, Murray y Wright (1981), pp. 107-15) mientras que en un método cuasi-Newton este requisito está, por construcción, garantizado (vid. por ejemplo Dennis y Schnabel (1983), pp. 198-211). El carácter definido positivo de G_k garantiza, por tanto, la existencia de una longitud de paso tal que el valor de la función objetivo decrece estrictamente de una iteración a otra. Sobre las posibles estrategias de cálculo de dicha longitud de paso, pueden consultarse Gill, Murray y Wright (1981), pp. 100-2 y Dennis y Schnabel (1983), pp. 116-29, cuyas sugerencias se han empleado en el algoritmo [MAXFV] (vid. Apéndice A.3).

[4] Para evaluar numéricamente las primeras derivadas de un campo escalar $F:E^n \rightarrow E$ diferenciable, pueden utilizarse dos aproximaciones. La primera de ellas ("*forward difference formula*") está dada por:

$$[\nabla F(x_k)]_i \approx \frac{F(x_k + h_i e_i) - F(x_k)}{h_i}, \quad i=1,2,\dots,n$$

donde e_i es el i -ésimo vector de la base canónica de E^n y h_i es una *perturbación* que, en términos matemáticos, tiende a cero. La fórmula anterior requiere tan sólo una evaluación adicional de la función objetivo, y supone un error implícito $O(h_i)$. La segunda aproximación ("*central difference formula*") está dada por:

$$[\nabla F(x_k)]_i \approx \frac{F(x_k + h_i e_i) - F(x_k - h_i e_i)}{2h_i}, \quad i=1,2,\dots,n$$

Esta aproximación requiere dos evaluaciones adicionales de la función objetivo por variable, pero el error implícito es tan sólo $O(h_i^2)$. Esta segunda fórmula es la que se emplea en el algoritmo [MAXFV] (vid. Apéndice A.3) y su utilización pone de manifiesto el interés en disponer de un método computacionalmente eficiente (rápido y preciso) para evaluar la función objetivo (la FVE en nuestro caso). Para más detalles (incluida la elección de los valores de h_i), puede consultarse Dennis y Schnabel (1983), pp. 77-80 y 103-6.

[5] Como es bien sabido [vid. por ejemplo Nicholls (1976, 1977); Anderson (1980)], para evaluar la precisión de las estimaciones puntuales (x^*) obtenidas por MVE, pueden utilizarse los elementos de la diagonal principal de la inversa de la matriz de información, evaluada en las estimaciones puntuales $[I(x^*)^{-1}]$, que es un estimador consistente de la matriz de covarianzas de la distribución (normal) asintótica del estimador por MVE. Teniendo en cuenta que, en el óptimo (x^*), las expresiones [3.1.1] (FVE) y [3.1.5] (FVE concentrada) coinciden, la matriz de información para un proceso ARMA multivariante puede escribirse como:

$$I(x) = E \left[-\frac{1}{2\Pi(x)^2} \nabla\Pi(x)\nabla\Pi(x)^T + \frac{n}{2\Pi(x)} H(x) \right]$$

donde $H(x)$ es la matriz hessiana de [3.1.8]. Una estimación de la matriz de covarianzas entre los parámetros puede obtenerse, a partir de esta expresión, ignorando el operador esperanza y teniendo en cuenta que $\nabla\Pi(x^*)=0$. Dicha estimación vendrá dada por:

$$COV(x^*) = \frac{2F(x^*)}{n} H(x^*)^{-1}$$

donde $H(x^*)$ es ahora la matriz hessiana de [3.1.18] evaluada en el óptimo (nótese que, en [3.1.18], $\Pi(x_0)=\Pi_{10}\times\Pi_{20}$ es una constante). La principal ventaja de trabajar con la expresión anterior, consiste en que su cálculo es inmediato a partir de la información acumulada en el curso de las iteraciones de un método cuasi-Newton. Básicamente, partiendo de $G_0 = |F(x_0)| \times I_k = I_k$ (matriz identidad de orden k , con $k = n^\circ$ de parámetros a estimar), en cada iteración de un método cuasi-Newton basado en la fórmula BFGS, se actualiza la información sobre la curvatura de la función objetivo (en nuestro caso, $F(x)$ en [3.1.18]) actualizando la matriz G_i , que determina la dirección de búsqueda (vid. [3.1.17]), mediante la siguiente fórmula:

$$G_{i+1} = G_i + \frac{y_i y_i^T}{\alpha_i y_i^T d_i} + \frac{\nabla F(x_i) \nabla F(x_i)^T}{\nabla F(x_i)^T d_i}$$

donde $y_i = \nabla F(x_{i+1}) - \nabla F(x_i)$. Cuando el algoritmo ha convergido, la última actualización, obtenida mediante la fórmula anterior, contiene la información suficiente sobre la curvatura de la función objetivo en el óptimo como para que la inversa de dicha actualización pueda considerarse una aproximación razonable a $\mathbf{H}(x^*)^{-1}$ (vid. Dennis y Schnabel (1983), pp. 206-7 y las referencias allí citadas). En cualquier caso, una comprobación de la validez de las covarianzas así obtenidas, puede llevarse a cabo como se describe al final del apartado 3.1.3 (la descripción detallada de todos estos cálculos puede encontrarse en el Apéndice A.3).

[6] El algoritmo [MAXFV] optimiza la función objetivo [3.1.18] respecto a los parámetros Φ_i ($i=1,2,\dots,p$), Θ_i ($i=1,2,\dots,q$), μ y \mathbf{Q} . Cuando el procedimiento ha convergido, el parámetro σ^2 que aparece en la FVE [3.1.1] se estima según [3.1.4] y, a la vista de [2.1.11], la estimación correspondiente de la matriz de covarianzas de \mathbf{a}_i se obtiene multiplicando por [3.1.4] cada componente de la matriz \mathbf{Q} estimada, tal y como se describe al final del apartado 3.1.1. En la práctica, se ha observado que esta estrategia por sí sola evita posibles problemas de escala cuando el orden de magnitud de las covarianzas de \mathbf{a}_i es bastante inferior al del resto de parámetros del modelo (Φ_i ($i=1,2,\dots,p$), Θ_i ($i=1,2,\dots,q$) y μ); como se ilustra en la estimación de [3.1.34], esta situación no es, ni mucho menos, improbable.

3.2. Otros procedimientos de estimacion

Como es bien sabido (vid. por ejemplo Box y Jenkins (1970), cap. 7; Ansley y Newbold (1980), pp. 159-61), para estimar los parámetros de un proceso ARMA, puede emplearse, además del criterio de máxima verosimilitud, el de *mínima suma de cuadrados* (MSC en adelante). Este criterio puede justificarse *per se* (minimizar las diferencias entre valores observados y ajustados) o partiendo de la expresión [3.1.9] e ignorando el término [3.1.11] (si el tamaño muestral n es lo suficientemente grande y el modelo está bien condicionado, el segundo factor de [3.1.11] puede considerarse despreciable). Entonces, teniendo en cuenta [2.3.36] y [2.2.23]-[2.2.26], es inmediato comprobar que la expresión [3.1.10] puede escribirse como:

$$(\eta^T \eta - \lambda^T \lambda) = \hat{\mathbf{u}}_*^T \mathbf{\Omega}^{-1} \hat{\mathbf{u}}_* + \hat{\mathbf{a}}^T (\mathbf{I}_n \otimes \mathbf{Q}^{-1}) \hat{\mathbf{a}} \quad [3.2.1]$$

donde $\hat{\mathbf{u}}_* = E[\mathbf{u}_* | \mathbf{w}]$ (vid. [2.3.41]) y $\hat{\mathbf{a}} = E[\mathbf{a} | \mathbf{w}]$ (vid. [2.3.42]). Por otro lado, a partir de las expresiones [2.1.16], [2.2.43] y [2.2.47] se deduce, para un proceso MA(q) puro ($p=0$), que:

$$\hat{\mathbf{u}}_*^T \boldsymbol{\Omega}^{-1} \hat{\mathbf{u}}_* = \sum_{t=1-q}^0 \hat{\mathbf{a}}_t^T \mathbf{Q}^{-1} \hat{\mathbf{a}}_t \quad [3.2.2]$$

Por tanto, dado que cualquier proceso ARMA(p, q) estacionario puede expresarse (aproximadamente) como un proceso MA puro de orden infinito, puede demostrarse que, cuando $p \neq 0$:

$$(\boldsymbol{\eta}^T \boldsymbol{\eta} - \boldsymbol{\lambda}^T \boldsymbol{\lambda}) = \hat{\mathbf{u}}_*^T \boldsymbol{\Omega}^{-1} \hat{\mathbf{u}}_* + \sum_{t=1}^n \hat{\mathbf{a}}_t^T \mathbf{Q}^{-1} \hat{\mathbf{a}}_t = \sum_{t=-\infty}^n \hat{\mathbf{a}}_t^T \mathbf{Q}^{-1} \hat{\mathbf{a}}_t \quad [3.2.3]$$

A partir de esta expresión, puede diseñarse, una primera versión del criterio *MSC* consistente en minimizar:

$$S(\boldsymbol{\Phi}_i, \boldsymbol{\Theta}_i, \boldsymbol{\mu}, \mathbf{Q} | \mathbf{w}) = \sum_{t=1-L}^n \hat{\mathbf{a}}_t^T \mathbf{Q}^{-1} \hat{\mathbf{a}}_t \quad [3.2.4]$$

En la práctica [3.2.4] se evalúa haciendo uso de la técnica de la *retrovisión* ("backcasting") de Box y Jenkins (1970), donde L es un número entero lo suficientemente grande como para que pueda considerarse que $\hat{\mathbf{a}}_t = \mathbf{0}$ para $t < 1-L$ (nótese que para un MA(q) puro ($p=0$), $L=q$; *vid.* [3.2.2]). Estrictamente hablando, la expresión [3.2.4] es tan sólo una aproximación a la suma de cuadrados *exacta* [3.2.3], ya que, numéricamente, L es un número finito y, aunque el cálculo de retrovisiones descrito por Box y Jenkins (1976) es un proceso iterativo, en la práctica suele llevarse a cabo en una sola iteración, probablemente por conveniencia computacional.

En particular, el estimador basado en minimizar [3.2.4] (*MSC con retrovisión*) es de dudosa validez cuando el módulo de alguna de las raíces de [2.1.6] y/o de [2.1.7] es cercano a la unidad y/o el tamaño muestral n es pequeño. En estas circunstancias, resulta necesario escoger L bastante grande y/o iterar varias veces en el cálculo de las retrovisiones [*vid.* por ejemplo Newbold (1974)], lo cual hace perder a este procedimiento su principal atractivo: la eficiencia computacional. Un procedimiento aún más rápido para obtener estimadores por *MSC* se deriva de igualar \mathbf{u}_* a su valor esperado de cero; la función a minimizar es en este caso:

$$S(\boldsymbol{\Phi}_i, \boldsymbol{\Theta}_i, \boldsymbol{\mu}, \mathbf{Q} | \mathbf{w}, \mathbf{u}_* = \mathbf{0}) = \sum_{t=1}^n \hat{\mathbf{a}}_{0t}^T \mathbf{Q}^{-1} \hat{\mathbf{a}}_{0t} \quad [3.2.5]$$

donde los $\hat{\mathbf{a}}_{0t}$ se calculan recursivamente según [2.3.2] (nótese que [3.2.5] se obtiene directamente de [3.1.9] ignorando el factor [3.1.11] y el segundo sumando de [3.1.10]).

Este procedimiento (*MSC condicional*) presenta problemas aún más graves que el procedimiento *MSC con retrovisión* cuando el proceso ARMA está cerca de o en la no invertibilidad y/o el tamaño muestral n es pequeño (*vid.* Ansley y Newbold (1980) para un análisis detallado de las propiedades en muestras pequeñas de estos dos estimadores en el contexto univariante). No obstante, puede demostrarse que las propiedades asintóticas de los estimadores por *MSC con retrovisión* y por *MSC condicional* son equivalentes a las del estimador por *MVE* [*vid.* por ejemplo Whittle (1953); Kabaila (1980, 1983)].

En el contexto univariante, las propiedades asintóticas y en muestras pequeñas de distintos criterios de estimación (máxima verosimilitud y mínima suma de cuadrados) y de las posibles versiones de los mismos (exacta, aproximada y condicional) son, por tanto, conocidas. Por ello, esta sección se limita a ilustrar la extensión de dichas propiedades al contexto multivariante, haciendo uso de las posibilidades que ofrece el mecanismo de estimación por *MVE* descrito en el Capítulo 2 y en la Sección 3.1. Probablemente, sería interesante realizar un análisis detallado de las propiedades en muestras pequeñas de los distintos estimadores disponibles para procesos ARMA multivariantes, aunque, en principio, no cabe esperar diferencias respecto a los resultados ya conocidos sobre procesos univariantes.

3.2.1. Estimación por máxima verosimilitud aproximada

En los casos en que el proceso ARMA considerado es claramente invertible, puede obtenerse un importante ahorro computacional si en el proceso de estimación se evalúa la función de verosimilitud haciendo uso de las propiedades descritas en el apartado 2.3.2 (*vid.* también Sección 2.4). Para contrastar empíricamente la validez del estimador por *MVA*, se ha llevado a cabo, en primer lugar, el mismo ejercicio de simulación que en el apartado 3.1.3 (un total de 1200 estimaciones por *MVA*, correspondientes a 100 simulaciones de cada uno de los 12 modelos derivados de [3.1.20]-[3.1.23]). Los resultados obtenidos, para un criterio δ de convergencia de las matrices \mathbf{Z}_i igual a 10^{-3} (*vid.* [2.3.44]), son *idénticos* a los resumidos en las Tablas 3.1.1-3.1.12 del apartado 3.1.3. Por supuesto, la diferencia entre ambos ejercicios se encuentra en el tiempo medio requerido por cada estimación (*vid.* Sección 3.3).

En segundo lugar, también se han estimado por *MVA* (con $\delta = 10^{-3}$) las representaciones bivariantes [3.1.24]-[3.1.25] y [3.1.26]-[3.1.27], cada una de ellas con 100 realizaciones independientes de tamaño $n=100$. En este caso, los resultados tampoco difieren significativamente de los obtenidos por *MVE*, aunque, debido, probablemente, a que ahora los modelos no son diagonales, los resultados no son idénticos. En concreto, para el proceso MA(2) [3.1.24]-[3.1.25] los resultados son:

$$\hat{\theta}_1 = \begin{bmatrix} -1.00443 & -1.80204 \\ (0.09689) & (0.06038) \\ 0.49521 & 1.00036 \\ (0.08533) & (0.05863) \end{bmatrix} \quad [3.2.6]$$

$$\hat{\theta}_2 = \begin{bmatrix} -0.81135 & -1.40314 \\ (0.17558) & (0.11874) \\ 0.40897 & 0.70185 \\ (0.14335) & (0.09400) \end{bmatrix} \quad [3.2.7]$$

$$E[\hat{a}_t \hat{a}_t^T] = \begin{bmatrix} 0.97807 \\ (0.12806) \\ 0.99948 & 1.96115 \\ (0.16319) & (0.29377) \end{bmatrix} \quad [3.2.8]$$

Y para el proceso ARMA(1,1) [3.1.26]-[3.1.27]:

$$\hat{\phi}_1 = \begin{bmatrix} 1.09883 & 0.51528 \\ (0.05413) & (0.04243) \\ -0.97644 & 0.38540 \\ (0.10147) & (0.07718) \end{bmatrix} \quad [3.2.9]$$

$$\hat{\theta}_1 = \begin{bmatrix} 1.00455 & 1.02254 \\ (0.08851) & (0.08497) \\ -0.77457 & -1.20280 \\ (0.11869) & (0.07624) \end{bmatrix} \quad [3.2.10]$$

$$E[\hat{a}_t \hat{a}_t^T] = \begin{bmatrix} 0.98152 \\ (0.13270) \\ 0.98478 & 1.93074 \\ (0.16770) & (0.29193) \end{bmatrix} \quad [3.2.11]$$

Comparando [3.2.6]-[3.2.11] con [3.1.28]-[3.1.33], es evidente que no existen diferencias significativas entre las estimaciones por *MVA* y por *MVE* [al igual que entonces, el error relativo medio en la estimación del proceso ARMA(1,1) es de un 2.0% y tan sólo de un 0.8% en la del MA(2)].

Por último, también se han estimado por *MVA* (con $\delta=10^{-3}$) las dos representaciones alternativas del modelo [3.1.34] consideradas al final del apartado 3.1.3. Los resultados obtenidos son *idénticos* a los que se resumen en la Tabla 3.1.13 de dicho apartado y en las ecuaciones [3.1.35] y [3.1.36] (por supuesto, también coinciden los residuos de las Figuras 3.1.1-3.1.8 y los perfiles de las Figuras 3.1.9-3.1.18). Para completar el análisis de dicho apartado, aprovechando la coincidencia de resultados, se ofrecen a continuación los perfiles de la *FVA* (que coinciden con los de la *FVE*) para la representación [3.1.34] obtenida eliminando el factor $(1-\alpha B)$ y reemplazando $m1_t=\nabla\log(M1_t)$ por $m1_t=\nabla^2\log(M1_t)$.

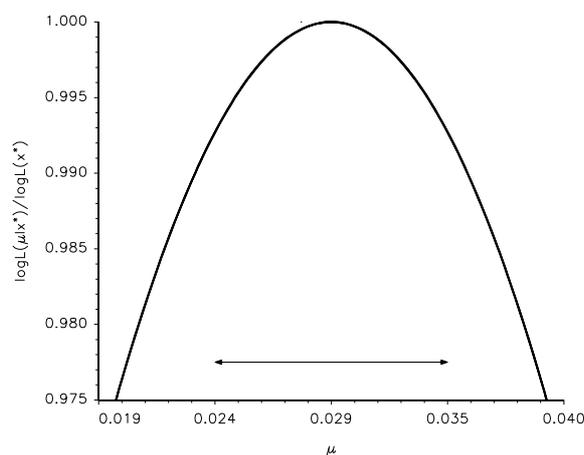


Fig. 3.2.1: Perfil para μ [0.029 (0.003)].

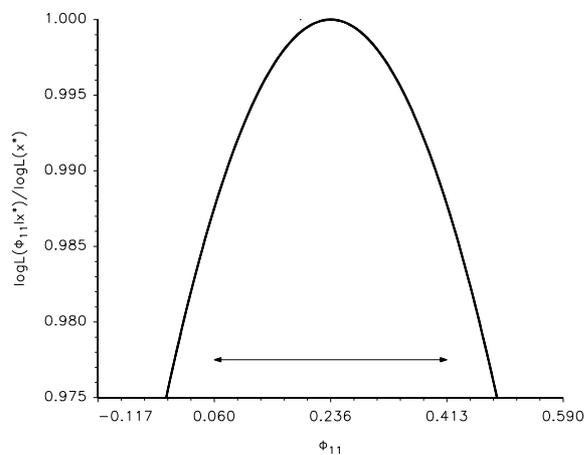


Fig. 3.2.2: Perfil para ϕ_{11} [0.236 (0.088)].

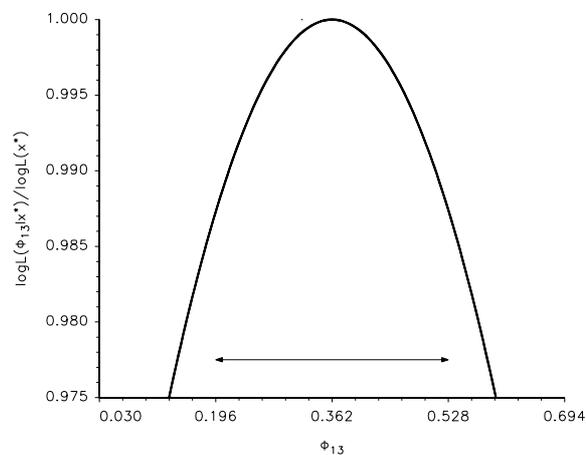


Fig. 3.2.3: Perfil para ϕ_{13} [0.362 (0.083)].

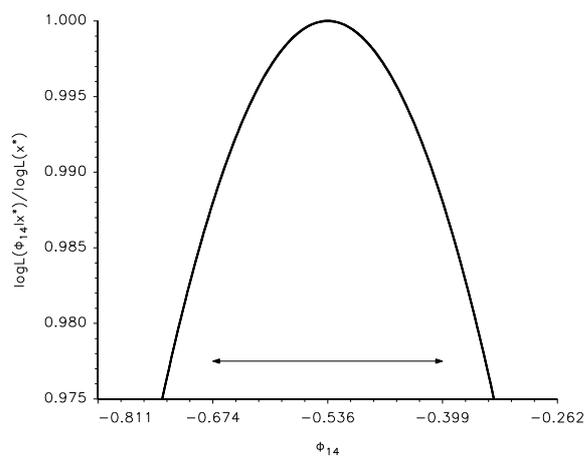


Fig. 3.2.4: Perfil para ϕ_{14} [-0.536 (0.069)].

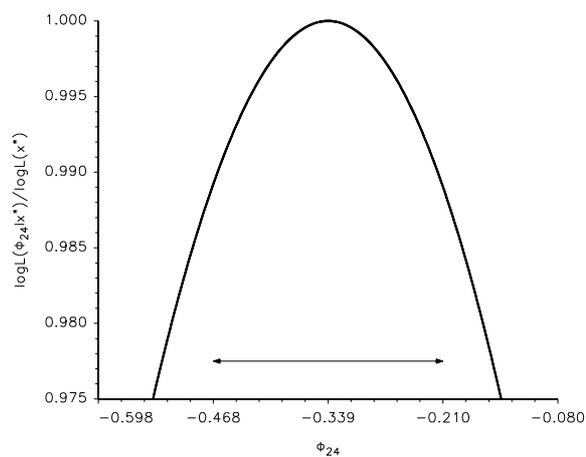


Fig. 3.2.5: Perfil para ϕ_{24} [-0.339 (0.065)].

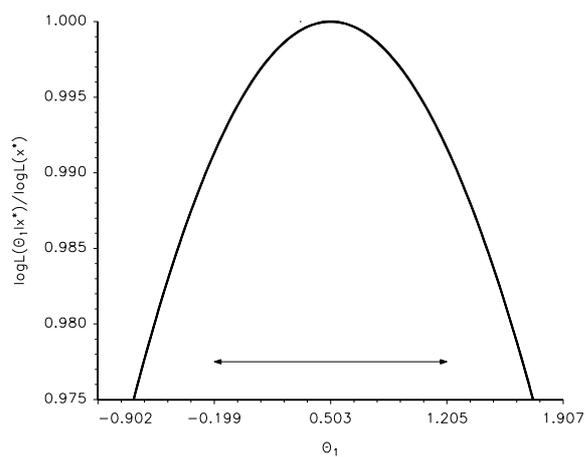


Fig. 3.2.6: Perfil para θ_1 [0.503 (0.351)].

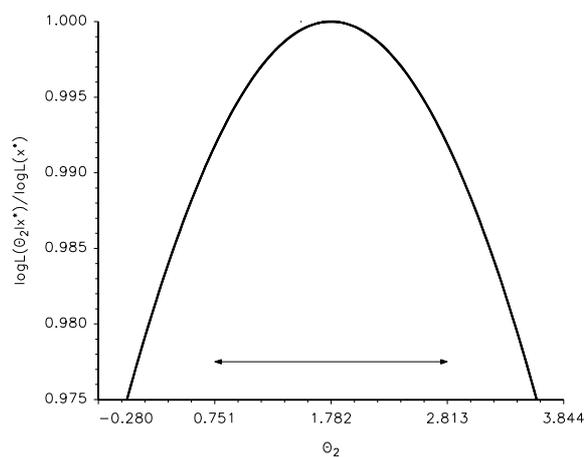


Fig. 3.2.7: Perfil para θ_2 [1.782 (0.515)].

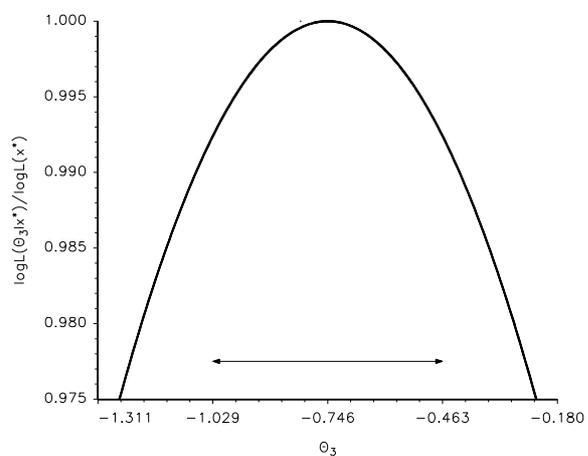


Fig. 3.2.8: Perfil para θ_3 [-0.746 (0.141)].

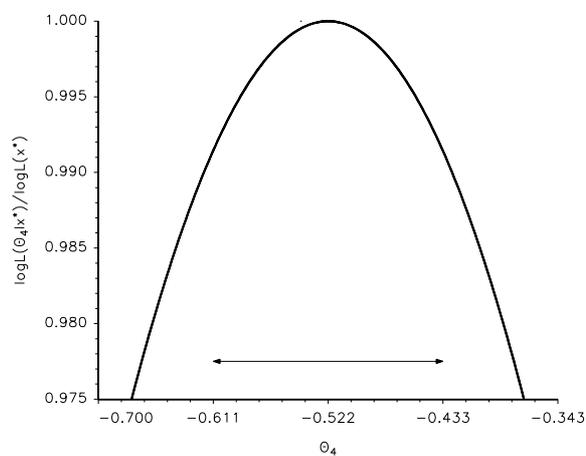


Fig. 3.2.9: Perfil para θ_4 [-0.522 (0.045)].

Al igual que en la estimación por *MVE* de la representación [3.1.34] completa, las Figuras 3.2.2-3.2.5 podrían indicar una ligera sobreestimación de los errores estándar correspondientes a los parámetros autorregresivos, que, en cualquier caso, resultan claramente significativos.

Sobre la base de todos estos resultados, puede concluirse, como era de esperar según lo expuesto sobre la *FVA* en el apartado 2.3.2 y en la Sección 2.4 del Capítulo 2, que el empleo de la *FVA* como función objetivo a maximizar en vez de la *FVE* no supone ninguna pérdida de fiabilidad de las estimaciones resultantes. Además, como el grado de precisión en la evaluación de la función de verosimilitud puede controlarse manipulando simplemente la tolerancia δ , se dispone de un procedimiento de estimación general, que incluye, como casos particulares, la estimación por *MVE* y por *MVA*, con la ganancia en tiempo de cálculo que puede suponer ésta sobre aquella (*vid.* Sección 2.4 y Sección 3.3).

3.2.2. Estimación por máxima verosimilitud condicional

Una última versión del criterio de estimación por máxima verosimilitud puede obtenerse, al igual que en el contexto del criterio *MSC*, igualando el vector \mathbf{u}_* de valores muestrales (*vid.* [2.1.16]) a su valor esperado de cero. Es inmediato comprobar que, en tal caso, la función de verosimilitud (*condicional* en $\mathbf{u}_* = \mathbf{0}$) resulta ser:

$$L(\Phi_i, \Theta_i, \mu, \sigma^2, \mathbf{Q} | \mathbf{w}, \mathbf{u}_* = \mathbf{0}) = (2\pi\sigma^2)^{-\frac{nm}{2}} \cdot |\mathbf{Q}|^{-\frac{n}{2}} \cdot \exp \left\{ -\frac{1}{2\sigma^2} \boldsymbol{\eta}^T \boldsymbol{\eta} \right\} \quad [3.2.12]$$

(compárese esta expresión con la *FVE* [3.1.1]). Concentrando la *FVC* [3.2.12] en el estimador por máxima verosimilitud de σ^2 (como al principio del apartado 3.1.1), resulta evidente que maximizar [3.2.12] es equivalente a minimizar la siguiente expresión:

$$|\mathbf{Q}|^{\frac{1}{m}} \times \boldsymbol{\eta}^T \boldsymbol{\eta} = |\mathbf{Q}|^{\frac{1}{m}} \times \sum_{t=1}^n \hat{\mathbf{a}}_{0t}^T \mathbf{Q}^{-1} \hat{\mathbf{a}}_{0t} \quad [3.2.13]$$

(*vid.* [3.2.5]). Para examinar las propiedades del estimador resultante de minimizar [3.2.13] (estimador por *MVC*), se ha efectuado, en primer lugar, el mismo ejercicio de simulación que en los apartados 3.1.3 (*MVE*) y 3.2.1 (*MVA*) (1200 estimaciones por *MVC*, correspondientes a 100 realizaciones de cada uno de los 12 modelos derivados de [3.1.20]-[3.1.23]). Los resultados se resumen en las Tablas 3.2.1-3.2.12.

M 1	ϕ_{11}	ϕ_{21}	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	0.95	0.40	1.00	1.00	2.00
<i>Valor medio estimado</i>	0.927 (0.044)	0.384 (0.091)	1.145 (0.325)	1.000 (0.262)	1.967 (0.428)

Tabla 3.2.1. Resultados de la estimación por MVC para un AR(1) bivalente diagonal ($n=50$; 100 repeticiones).

M 2	ϕ_{11}	ϕ_{12}	ϕ_{21}	ϕ_{22}	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	0.95	-0.96	0.85	-0.95	1.00	1.00	2.00
<i>Valor medio estimado</i>	0.939 (0.053)	-0.939 (0.046)	0.825 (0.054)	-0.920 (0.054)	1.510 (0.756)	1.190 (0.608)	2.864 (1.285)

Tabla 3.2.2. Resultados de la estimación por MVC para un AR(2) bivalente diagonal ($n=50$; 100 repeticiones).

M 3	θ_{11}	θ_{21}	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	0.30	0.70	1.00	1.00	2.00
<i>Valor medio estimado</i>	0.300 (0.120)	0.691 (0.077)	1.002 (0.194)	0.987 (0.246)	1.982 (0.435)

Tabla 3.2.3. Resultados de la estimación por MVC para un MA(1) bivalente diagonal ($n=50$; 100 repeticiones).

M 4	θ_{11}	θ_{12}	θ_{21}	θ_{22}	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	0.95	-0.96	0.85	-0.95	1.00	1.00	2.00
<i>Valor medio estimado</i>	0.891 (0.092)	-0.853 (0.092)	0.820 (0.091)	-0.840 (0.112)	1.153 (0.241)	1.143 (0.284)	2.247 (0.533)

Tabla 3.2.4. Resultados de la estimación por MVC para un MA(2) bivalente diagonal ($n=50$; 100 repeticiones).

M 5	ϕ_{11}	ϕ_{21}	θ_{11}	θ_{21}	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	0.90	0.30	0.70	0.30	1.00	1.00	2.00
<i>Valor medio estimado</i>	0.823 (0.178)	0.200 (0.515)	0.621 (0.232)	0.195 (0.541)	1.000 (0.197)	0.984 (0.247)	1.942 (0.427)

Tabla 3.2.5. Resultados de la estimación por MVC para un ARMA(1,1) bivalente diagonal ($n=50$; 100 repeticiones).

M 5	ϕ_{11}	ϕ_{21}	θ_{11}	θ_{21}	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	-0.90	-0.80	0.40	0.50	1.00	1.00	2.00
<i>Valor medio estimado</i>	-0.875 (0.066)	-0.766 (0.073)	0.358 (0.123)	0.464 (0.110)	1.172 (0.311)	1.110 (0.306)	2.181 (0.538)

Tabla 3.2.6. Resultados de la estimación por MVC para un ARMA(1,1) bivalente diagonal ($n=50$; 100 repeticiones).

M 15	ϕ_{11}	ϕ_{21}	Φ_1	Φ_2	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	0.20	0.40	0.90	0.40	1.00	1.00	2.00
<i>Valor medio estimado</i>	0.191	0.396	0.884	0.391	1.286	1.005	1.970
<i>Error estándar estimado</i>	(0.143)	(0.114)	(0.058)	(0.103)	(0.291)	(0.280)	(0.432)

Tabla 3.2.7. Resultados de la estimación por MVC para un $AR(1) \times AR(1)_4$ bivalente diagonal ($n=50$; 100 repeticiones).

M 18	ϕ_{11}	ϕ_{21}	Φ_1	Φ_2	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	0.90	0.40	0.20	0.95	1.00	1.00	2.00
<i>Valor medio estimado</i>	0.877	0.366	0.184	0.944	1.024	1.025	4.107
<i>Error estándar estimado</i>	(0.045)	(0.135)	(0.092)	(0.028)	(0.148)	(0.239)	(0.907)

Tabla 3.2.8. Resultados de la estimación por MVC para un $AR(1) \times AR(1)_{12}$ bivalente diagonal ($n=100$; 100 repeticiones).

M 22	θ_{11}	θ_{21}	Θ_1	Θ_2	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	0.60	0.80	0.70	0.90	1.00	1.00	2.00
<i>Valor medio estimado</i>	0.547	0.681	0.610	0.714	1.083	1.148	2.542
<i>Error estándar estimado</i>	(0.120)	(0.125)	(0.113)	(0.101)	(0.218)	(0.308)	(0.649)

Tabla 3.2.9. Resultados de la estimación por MVC para un $MA(1) \times MA(1)_4$ bivalente diagonal ($n=50$; 100 repeticiones).

M 25	θ_{11}	θ_{21}	θ_1	θ_2	σ_{11}^2	σ_{12}^2	σ_{22}^2
Valor del parámetro	0.90	0.70	0.90	0.80	1.00	1.00	2.00
Valor medio estimado	0.841 (0.064)	0.673 (0.062)	0.675 (0.071)	0.647 (0.069)	1.325 (0.211)	1.213 (0.214)	2.300 (0.351)

Tabla 3.2.10. Resultados de la estimación por MVC para un $MA(1) \times MA(1)_{12}$ bivariente diagonal ($n=100$; 100 repeticiones).

M 11	ϕ_1	ϕ_2	θ_1	θ_2	σ_{11}^2	σ_{12}^2	σ_{22}^2
Valor del parámetro	-0.90	-0.80	0.40	0.50	1.00	1.00	2.00
Valor medio estimado	-0.881 (0.066)	-0.769 (0.096)	0.244 (0.102)	0.360 (0.105)	1.816 (0.610)	1.562 (0.536)	2.755 (0.784)

Tabla 3.2.11. Resultados de la estimación por MVC para un $ARMA(1,1)_4$ bivariente diagonal ($n=50$; 100 repeticiones).

M 14	ϕ_1	ϕ_2	θ_1	θ_2	σ_{11}^2	σ_{12}^2	σ_{22}^2
Valor del parámetro	-0.90	-0.80	0.40	0.50	1.00	1.00	2.00
Valor medio estimado	-0.898 (0.040)	-0.803 (0.050)	0.186 (0.071)	0.267 (0.083)	2.193 (0.470)	1.818 (0.426)	3.306 (0.636)

Tabla 3.2.12. Resultados de la estimación por MVC para un $ARMA(1,1)_{12}$ bivariente diagonal ($n=100$; 100 repeticiones).

Según estos resultados, pueden hacerse las siguientes observaciones: (i) tanto en procesos AR puros como en procesos mixtos, las estimaciones de los parámetros autorregresivos son bastante precisas; (ii) no obstante, las estimaciones de los parámetros MA sólo son precisas cuando no hay parte AR y, además, el proceso es claramente invertible; (iii) en el caso de procesos mixtos, los parámetros MA están siempre infraestimados (aunque el proceso sea claramente invertible), tanto más cuanto más cerca están dichos parámetros de la no invertibilidad (esta última matización también es cierta para el caso de procesos MA puros); (iv) las estimaciones de las covarianzas del vector de residuos son, prácticamente en todos los casos, de dudosa validez, en especial si el orden del proceso es elevado o los parámetros MA están infraestimados (cuando están cercanos a la no invertibilidad o en el caso de procesos mixtos).

En segundo lugar, también se han estimado por *MVC* las representaciones bivariantes [3.1.24]-[3.1.25] y [3.1.26]-[3.1.27], cada una de ellas con 100 realizaciones independientes de tamaño $n=100$. Para el proceso MA(2) [3.1.24]-[3.1.25] los resultados (medias y errores estándar muestrales) de las 100 estimaciones por *MVC* son los siguientes:

$$\hat{\theta}_1 = \begin{pmatrix} -1.06436 & -1.76612 \\ (0.13596) & (0.08863) \\ 0.53437 & 0.97410 \\ (0.08800) & (0.06567) \end{pmatrix} \quad [3.2.14]$$

$$\hat{\theta}_2 = \begin{pmatrix} -0.80281 & -1.41293 \\ (0.13732) & (0.10181) \\ 0.40595 & 0.70775 \\ (0.11209) & (0.07903) \end{pmatrix} \quad [3.2.15]$$

$$E[\hat{a}_i \hat{a}_i^T] = \begin{pmatrix} 1.16385 & \\ (0.27423) & \\ 0.89940 & 2.01798 \\ (0.21726) & (0.31977) \end{pmatrix} \quad [3.2.16]$$

Y para el proceso ARMA(1,1) [3.1.26]-[3.1.27]:

$$\hat{\Phi}_1 = \begin{pmatrix} 1.09819 & 0.51648 \\ (0.05660) & (0.04202) \\ -0.97440 & 0.38473 \\ (0.11406) & (0.07496) \end{pmatrix} \quad [3.2.17]$$

$$\hat{\Theta}_1 = \begin{pmatrix} 1.02303 & 0.92862 \\ (0.01780) & (0.15723) \\ -0.93498 & -1.00358 \\ (0.20814) & (0.17697) \end{pmatrix} \quad [3.2.18]$$

$$E[\hat{a}_t \hat{a}_t^T] = \begin{pmatrix} 1.37578 \\ (0.43956) \\ 0.62649 & 2.61557 \\ (0.35157) & (0.91577) \end{pmatrix} \quad [3.2.19]$$

El error relativo medio en la estimación del proceso MA(2) es de un 4.5% (frente al 0.8% correspondiente a las estimaciones por *MVE* y *MVA*), mientras que el cometido en la estimación del proceso ARMA(1,1) alcanza el 14.4% (frente al 2.0% en las estimaciones por *MVE* y *MVA*). En particular, el error relativo medio en la estimación de las covarianzas de los residuos es de un 9.1% para el proceso MA(2) y de un 35.3% para el proceso ARMA(1,1) (frente al 1.4% y al 2.3%, respectivamente, obtenidos por *MVE* y *MVA*).

Por último, también se han estimado por *MVC* las dos representaciones alternativas del modelo [3.1.34] consideradas al final de los apartados 3.1.3 y 3.2.1. Los resultados obtenidos se resumen en la Tabla 3.2.13.

	μ	ϕ_{11}	ϕ_{13}	ϕ_{14}	ϕ_{24}	α	θ_1	θ_2	θ_3	θ_4
Estimaciones con $\alpha=1$	0.032 (0.003)	0.281 (0.089)	0.395 (0.105)	-0.496 (0.072)	-0.312 (0.071)	* *	0.435 (0.306)	1.615 (0.571)	-0.689 (0.168)	-0.506 (0.055)
Estimaciones con α libre	0.029 (0.003)	0.283 (0.085)	0.463 (0.103)	-0.506 (0.068)	-0.316 (0.067)	1.012 (0.011)	0.364 (0.356)	1.576 (0.586)	-0.753 (0.150)	-0.511 (0.053)

Tabla 3.2.13. Estimaciones por *MVC* del modelo [3.1.34] con y sin la restricción $\alpha=1$ (un * indica que el parámetro correspondiente no se ha estimado).

La estimación de la matriz de covarianzas de los residuos, correspondiente al modelo estimado por *MVC* con $\alpha=1$, es la siguiente:

$$E[\hat{\mathbf{a}}_t \hat{\mathbf{a}}_t^T] = \begin{pmatrix} 0.57 \times 10^{-3} & & & \\ 0.58 \times 10^{-4} & 0.20 \times 10^{-3} & & \\ -0.80 \times 10^{-4} & 0.44 \times 10^{-5} & 0.40 \times 10^{-4} & \\ 0.42 \times 10^{-3} & 0.43 \times 10^{-4} & -0.51 \times 10^{-4} & 0.36 \times 10^{-3} \end{pmatrix} \quad [3.2.20]$$

mientras que la correspondiente a la estimación por *MVC* que no impone $\alpha=1$ es:

$$E[\hat{\mathbf{a}}_t \hat{\mathbf{a}}_t^T] = \begin{pmatrix} 0.58 \times 10^{-3} & & & \\ 0.89 \times 10^{-4} & 0.23 \times 10^{-3} & & \\ -0.62 \times 10^{-4} & 0.49 \times 10^{-5} & 0.33 \times 10^{-4} & \\ 0.44 \times 10^{-3} & 0.52 \times 10^{-4} & -0.43 \times 10^{-4} & 0.40 \times 10^{-3} \end{pmatrix} \quad [3.2.21]$$

Es interesante observar que el parámetro α estimado es *mayor* que la unidad. En el contexto de la estimación por *MVC* es perfectamente posible que las estimaciones puntuales de algunos parámetros estén fuera de las regiones de estacionariedad o de invertibilidad, ya que, como puede observarse en [3.2.12] y [3.2.13], el único requisito para evaluar la *FVC* consiste en que la matriz \mathbf{Q} sea definida positiva (sobre esta cuestión, pueden encontrarse más detalles en Ansley y Newbold (1980), pp. 162-3).

Por lo demás, es evidente que, en conjunto, las estimaciones obtenidas por *MVC* (Tabla 3.2.13) *no* pueden considerarse significativamente distintas de las obtenidas por *MVE* y *MVA* (Tabla 3.1.13). Esta afirmación puede contrastarse, si se desea, con carácter *más formal*, haciendo uso del estadístico [3.1.19], cuyo valor, considerando la hipótesis nula de igualdad entre las estimaciones obtenidas por *MVC* y *MVE* (o *MVA*) es $\xi_{RV}=5.1$ (el valor de la *FVE* evaluada en las estimaciones por *MVC* es igual a 366.006).

En resumen, el mecanismo de estimación por *MVC* puede proporcionar estimaciones *fiabiles* en situaciones bien condicionadas (como se muestra en el último ejemplo considerado). No obstante, en determinados casos (sobre todo, cuando algunos parámetros están cercanos a la no invertibilidad), la validez de las estimaciones obtenidas por *MVC* es más que cuestionable. Sin embargo, dada la sencillez y la eficiencia computacional (*vid.* Sección 3.3) de este procedimiento, su utilización puede resultar interesante en los primeros pasos de modelización de un proceso ARMA (*vid.* Tiao y Box (1981), pp.

809) y en la obtención de estimaciones preliminares de los parámetros, que sirvan de punto de partida para un algoritmo de estimación por *MVE* o *MVA* [sobre la obtención de estimaciones preliminares de los parámetros, pueden consultarse Shea (1987); Koreisha y Pukkila (1989)].

3.3. Análisis comparativo de distintos procedimientos de estimacion

Sobre la base de los resultados obtenidos en los apartados 3.1.3 (*FVE*), 3.2.1 (*FVA*) y 3.2.2 (*FVC*) pueden realizarse los siguientes comentarios:

- [1] En todos los casos analizados, las estimaciones obtenidas por *MVE*, utilizando el algoritmo [MAXFV], son adecuadas, incluso con los tamaños muestrales considerados y con algunos de los parámetros cerca de no estacionariedad o no invertibilidad.
- [2] En ninguno de los casos analizados, las estimaciones obtenidas por *MVA* difieren significativamente de las obtenidas por *MVE*.
- [3] Las estimaciones por *MVC* de modelos bien condicionados suelen ser aceptables. No obstante, en determinados casos, las estimaciones de los parámetros MA y de las covarianzas entre los residuos no pueden considerarse adecuadas, sobre todo cuando los parámetros MA están cerca de no invertibilidad. En estos casos, los resultados obtenidos por *MVE* o por *MVA* son claramente superiores a los obtenidos por *MVC*.

Para finalizar la comparación entre las tres versiones del criterio de estimación por máxima verosimilitud consideradas en este capítulo, en la Tabla 3.3.1 se recogen los tiempos empleados por el algoritmo [MAXFV] en la estimación por *MVE*, *MVA* y *MVC* de todos los ejemplos examinados. En el caso de los 14 modelos de [3.1.20]-[3.1.23] y [3.1.24]-[3.1.27], los tiempos que se presentan son tiempos *medios* (la suma de los tiempos empleados en cada una de las 100 estimaciones dividida entre 100), mientras que para las dos representaciones alternativas de [3.1.34], los tiempos son los de la única estimación llevada a cabo. El código empleado se ha compilado con *Turbo Pascal* 6.0 y se ha ejecutado en un ordenador personal *HP Vectra* 386/25 con un coprocesador matemático *Intel* 387DX-25. Por supuesto, los tiempos variarán si se utilizan otros lenguajes, compiladores y ordenadores.

	<i>MVE</i>	<i>MVA</i>	<i>MVC</i>
AR(1) bivalente diagonal	6"	6"	4"
AR(2) bivalente diagonal	19"	19"	10"
MA(1) bivalente diagonal	8"	6"	3"
MA(2) bivalente diagonal	39"	39"	15"
ARMA(1,1) bivalente diagonal sobreparametrizado	35"	27"	13"
ARMA(1,1) bivalente diagonal	21"	15"	8"
AR(1)×AR(1) ₄ bivalente diagonal	1' 5"	1' 5"	13"
AR(1)×AR(1) ₁₂ bivalente diagonal	11' 40"	11' 40"	51"
MA(1)×MA(1) ₄ bivalente diagonal	1' 22"	1' 22"	24"
MA(1)×MA(1) ₁₂ bivalente diagonal	7' 31"	7' 31"	1' 25"
ARMA(1,1) ₄ bivalente diagonal	1' 39"	1' 4"	18"
ARMA(1,1) ₁₂ bivalente diagonal	11' 24"	10' 27"	1' 20"
MA(2) bivalente	3'	1' 20"	54"
ARMA(1,1) bivalente	3' 2"	1' 57"	1' 14"
Modelo [3.1.34] sin $(1-\alpha)$ y con $m_{1t}=\nabla^2\log(M1_t)$	1h 18' 33"	1h 6' 13"	6' 37"
Modelo [3.1.34] completo	4h 48' 42"	3h 18' 28"	13' 54"

Tabla 3.3.1. Tiempos empleados en la estimación de los modelos considerados en este capítulo.

En contra de los defectos mencionados anteriormente, la estimación por *MVC* es, con diferencia, el procedimiento computacionalmente más rápido. Por otro lado, en consonancia con lo expuesto en el Capítulo 2, la estimación por *MVA* es significativamente más rápida que la estimación por *MVE*, excepto para procesos AR puros y para procesos con parte MA cerca de no invertibilidad. Es importante subrayar que el tiempo de estimación de cualquier proceso es altamente dependiente de (i) el número de parámetros a estimar, (ii) el tamaño de la muestra y (iii) la proximidad de los parámetros AR y MA a las fronteras de sus conjuntos admisibles. Por tanto, los resultados anteriores tan sólo deben considerarse como una ilustración de la carga computacional implícita en la estimación de procesos ARMA multivariantes. Dichos resultados pueden variar sensiblemente con tan sólo un cambio en los valores teóricos de los parámetros o en el tamaño de las muestras consideradas.

Para concluir esta sección, se han realizado con la versión *exacta* del procedimiento de Hillmer y Tiao (1979), ofrecida por *The S.C.A. Statistical System* [vid. Liu y Hudak (1986)], los mismos ejercicios de estimación que en las secciones anteriores. Los resultados correspondientes a las 100 estimaciones de cada uno de los 12 modelos derivados de [3.1.20]-[3.1.23] se resumen en las Tablas 3.3.2-3.3.13.

M 1	ϕ_{11}	ϕ_{21}	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	0.95	0.40	1.00	1.00	2.00
<i>Valor medio estimado</i>	0.927 (0.044)	0.384 (0.091)	1.007 (0.200)	0.992 (0.250)	1.961 (0.431)

Tabla 3.3.2. Resultados de la estimación con S.C.A. para un AR(1) bivariente diagonal ($n=50$; 100 repeticiones).

M 2	ϕ_{11}	ϕ_{12}	ϕ_{21}	ϕ_{22}	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	0.95	-0.96	0.85	-0.95	1.00	1.00	2.00
<i>Valor medio estimado</i>	0.943 (0.048)	-0.942 (0.044)	0.839 (0.040)	-0.927 (0.049)	0.992 (0.203)	0.986 (0.261)	1.929 (0.432)

Tabla 3.3.3. Resultados de la estimación con S.C.A. para un AR(2) bivariente diagonal ($n=50$; 100 repeticiones).

M 3	θ_{11}	θ_{21}	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	0.30	0.70	1.00	1.00	2.00
<i>Valor medio estimado</i>	0.306 (0.123)	0.704 (0.082)	1.005 (0.197)	0.993 (0.250)	1.957 (0.429)

Tabla 3.3.4. Resultados de la estimación con S.C.A. para un MA(1) bivariente diagonal ($n=50$; 100 repeticiones).

M 4	θ_{11}	θ_{12}	θ_{21}	θ_{22}	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	0.95	-0.96	0.85	-0.95	1.00	1.00	2.00
<i>Valor medio estimado</i>	0.946	-0.951	0.850	-0.948	1.002	1.002	1.953
<i>Error estándar estimado</i>	(0.066)	(0.050)	(0.063)	(0.063)	(0.200)	(0.243)	(0.432)

Tabla 3.3.5. Resultados de la estimación con S.C.A. para un MA(2) bivariante diagonal ($n=50$; 100 repeticiones).

M 5	ϕ_{11}	ϕ_{21}	θ_{11}	θ_{21}	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	0.90	0.30	0.70	0.30	1.00	1.00	2.00
<i>Valor medio estimado</i>	0.769	0.172	0.579	0.186	0.995	0.995	1.936
<i>Error estándar estimado</i>	(0.234)	(0.405)	(0.277)	(0.456)	(0.201)	(0.257)	(0.437)

Tabla 3.3.6. Resultados de la estimación con S.C.A. para un ARMA(1,1) bivariante diagonal ($n=50$; 100 repeticiones).

M 5	ϕ_{11}	ϕ_{21}	θ_{11}	θ_{21}	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	-0.90	-0.80	0.40	0.50	1.00	1.00	2.00
<i>Valor medio estimado</i>	-0.877	-0.771	0.414	0.515	0.997	0.989	1.942
<i>Error estándar estimado</i>	(0.066)	(0.073)	(0.129)	(0.107)	(0.203)	(0.255)	(0.437)

Tabla 3.3.7. Resultados de la estimación con S.C.A. para un ARMA(1,1) bivariante diagonal ($n=50$; 100 repeticiones).

M 15	ϕ_{11}	ϕ_{21}	Φ_1	Φ_2	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	0.20	0.40	0.90	0.40	1.00	1.00	2.00
<i>Valor medio estimado</i>	0.201	0.397	0.884	0.387	0.995	0.989	1.936
<i>Error estándar estimado</i>	(0.117)	(0.108)	(0.057)	(0.104)	(0.204)	(0.263)	(0.446)

Tabla 3.3.8. Resultados de la estimación con S.C.A. para un $AR(1) \times AR(1)_4$ bivariente diagonal ($n=50$; 100 repeticiones).

M 18	ϕ_{11}	ϕ_{21}	Φ_1	Φ_2	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	0.90	0.40	0.20	0.95	1.00	1.00	2.00
<i>Valor medio estimado</i>	0.881	0.383	0.190	0.945	0.993	0.996	1.961
<i>Error estándar estimado</i>	(0.044)	(0.069)	(0.088)	(0.026)	(0.166)	(0.191)	(0.358)

Tabla 3.3.9. Resultados de la estimación con S.C.A. para un $AR(1) \times AR(1)_{12}$ bivariente diagonal ($n=100$; 100 repeticiones).

M 22	θ_{11}	θ_{21}	Θ_1	Θ_2	σ_{11}^2	σ_{12}^2	σ_{22}^2
<i>Valor del parámetro</i>	0.60	0.80	0.70	0.90	1.00	1.00	2.00
<i>Valor medio estimado</i>	0.608	0.800	0.710	0.912	1.000	1.016	1.940
<i>Error estándar estimado</i>	(0.119)	(0.086)	(0.123)	(0.098)	(0.201)	(0.259)	(0.430)

Tabla 3.3.10. Resultados de la estimación con S.C.A. para un $MA(1) \times MA(1)_4$ bivariente diagonal ($n=50$; 100 repeticiones).

M 25	θ_{11}	θ_{21}	θ_1	θ_2	σ_{11}^2	σ_{12}^2	σ_{22}^2
Valor del parámetro	0.90	0.70	0.90	0.80	1.00	1.00	2.00
Valor medio estimado	0.920 (0.053)	0.715 (0.060)	0.901 (0.104)	0.829 (0.115)	0.962 (0.217)	0.958 (0.237)	1.853 (0.445)

Tabla 3.3.11. Resultados de la estimación con S.C.A. para un $MA(1) \times MA(1)_{12}$ bivalente diagonal ($n=100$; 100 repeticiones).

M 11	ϕ_1	ϕ_2	θ_1	θ_2	σ_{11}^2	σ_{12}^2	σ_{22}^2
Valor del parámetro	-0.90	-0.80	0.40	0.50	1.00	1.00	2.00
Valor medio estimado	-0.887 (0.059)	-0.786 (0.082)	0.400 (0.156)	0.514 (0.132)	0.994 (0.211)	0.987 (0.264)	1.913 (0.432)

Tabla 3.3.12. Resultados de la estimación con S.C.A. para un $ARMA(1,1)_4$ bivalente diagonal ($n=50$; 100 repeticiones).

M 14	ϕ_1	ϕ_2	θ_1	θ_2	σ_{11}^2	σ_{12}^2	σ_{22}^2
Valor del parámetro	-0.90	-0.80	0.40	0.50	1.00	1.00	2.00
Valor medio estimado	-0.905 (0.035)	-0.813 (0.044)	0.399 (0.104)	0.495 (0.105)	0.945 (0.275)	0.940 (0.290)	1.840 (0.556)

Tabla 3.3.13. Resultados de la estimación con S.C.A. para un $ARMA(1,1)_{12}$ bivalente diagonal ($n=100$; 100 repeticiones).

Si bien, en conjunto, estas estimaciones no pueden considerarse significativamente distintas de las obtenidas por *MVE* (o *MVA*) con los algoritmos **[MAXFV]** y **[FVE]** (*vid.* Tablas 3.1.1-3.1.12), es interesante señalar lo siguiente: (i) con muy pocas excepciones, los errores estándar de las estimaciones obtenidas con *S.C.A.* son ligeramente mayores que los obtenidos por *MVE*, lo cual parece reflejar una mayor eficiencia del algoritmo **[MAXFV]**, debida a la estimación simultánea de todos los parámetros; (ii) la estimación proporcionada por *S.C.A.* del modelo sobreparametrizado de la Tabla 3.3.6, es claramente inferior incluso a la obtenida por *MVC* (*vid.* Tabla 3.2.5); (iii) las estimaciones de la matriz de covarianzas de los residuos en las Tablas 3.3.11 y 3.3.13 son ligeramente inferiores a las correspondientes de las Tablas 3.1.10 y 3.1.12.

Esta última circunstancia se puede observar más claramente en las estimaciones obtenidas con *S.C.A.* de las representaciones bivariantes [3.1.24]-[3.1.25] y [3.1.26]-[3.1.27]. Para el proceso MA(2) [3.1.24]-[3.1.25] los resultados (medias y errores estándar muestrales) de las 100 estimaciones con *S.C.A.* son los siguientes:

$$\hat{\theta}_1 = \begin{bmatrix} -1.0118 & -1.7974 \\ (0.0938) & (0.0586) \\ 0.4975 & 0.9983 \\ (0.0862) & (0.0599) \end{bmatrix} \quad [3.3.1]$$

$$\hat{\theta}_2 = \begin{bmatrix} -0.8158 & -1.4006 \\ (0.1761) & (0.1208) \\ 0.4161 & 0.6947 \\ (0.1441) & (0.0963) \end{bmatrix} \quad [3.3.2]$$

$$E[\hat{a}_t \hat{a}_t^T] = \begin{bmatrix} 0.9256 & \\ (0.2680) & \\ 0.9616 & 1.8765 \\ (0.2912) & (0.5558) \end{bmatrix} \quad [3.3.3]$$

Y para el proceso ARMA(1,1) [3.1.26]-[3.1.27]:

$$\hat{\Phi}_1 = \begin{bmatrix} 1.0985 & 0.5157 \\ (0.0532) & (0.0411) \\ -0.9759 & 0.3849 \\ (0.1042) & (0.0731) \end{bmatrix} \quad [3.3.4]$$

$$\hat{\Theta}_1 = \begin{bmatrix} 1.0026 & 1.0240 \\ (0.0898) & (0.0856) \\ -0.7733 & -1.2024 \\ (0.1187) & (0.0736) \end{bmatrix} \quad [3.3.5]$$

$$E[\hat{a}_t \hat{a}_t^T] = \begin{bmatrix} 0.9267 & \\ (0.2669) & \\ 0.9381 & 1.8325 \\ (0.2873) & (0.5476) \end{bmatrix} \quad [3.3.6]$$

El error relativo medio en la estimación del proceso MA(2) es de un 2.4% (frente al 0.8% correspondiente a las estimaciones por *MVE*), mientras que el cometido en la estimación del proceso ARMA(1,1) alcanza el 3.4% (frente al 2.0% en las estimaciones por *MVE*). En particular, el error relativo medio en la estimación de las covarianzas de los residuos es de un 5.8% para el proceso MA(2) y de un 7.3% para el proceso ARMA(1,1) (frente al 1.4% y al 2.3%, respectivamente, obtenidos por *MVE* con los algoritmos [MAXFV] y [FVE]).

Por último, también se han estimado con *S.C.A.* las dos representaciones alternativas del modelo [3.1.34] consideradas al final de los apartados 3.1.3 y 3.2.1. Los resultados obtenidos se resumen en la Tabla 3.3.14.

	μ	ϕ_{11}	ϕ_{13}	ϕ_{14}	ϕ_{24}	α	θ_1	θ_2	θ_3	θ_4
Estimaciones con $\alpha=1$	0.028 (0.002)	0.232 (0.086)	0.357 (0.080)	-0.551 (0.069)	-0.354 (0.062)	* *	0.525 (0.347)	1.749 (0.480)	-0.753 (0.136)	-0.562 (0.047)
Estimaciones con α libre	0.026 (0.003)	0.267 (0.089)	0.313 (0.081)	-0.584 (0.073)	-0.401 (0.064)	1.003 (0.009)	0.543 (0.344)	1.465 (0.476)	-0.791 (0.137)	-0.482 (0.045)

Tabla 3.3.14. Estimaciones con *S.C.A.* del modelo [3.1.34] con y sin la restricción $\alpha=1$ (un * indica que el parámetro correspondiente no se ha estimado).

La estimación de la matriz de covarianzas de los residuos, correspondiente al modelo estimado con *S.C.A.* con $\alpha=1$, es la siguiente:

$$E[\hat{\mathbf{a}}_t \hat{\mathbf{a}}_t^T] = \begin{pmatrix} 0.60 \times 10^{-3} & & & \\ 0.11 \times 10^{-3} & 0.17 \times 10^{-3} & & \\ -0.73 \times 10^{-4} & -0.20 \times 10^{-4} & 0.32 \times 10^{-4} & \\ 0.44 \times 10^{-3} & 0.94 \times 10^{-4} & -0.40 \times 10^{-4} & 0.38 \times 10^{-3} \end{pmatrix} \quad [3.3.7]$$

mientras que la correspondiente a la estimación con *S.C.A.* que no impone $\alpha=1$ es:

$$E[\hat{\mathbf{a}}_t \hat{\mathbf{a}}_t^T] = \begin{pmatrix} 0.60 \times 10^{-3} & & & \\ 0.13 \times 10^{-3} & 0.20 \times 10^{-3} & & \\ -0.76 \times 10^{-4} & -0.16 \times 10^{-4} & 0.34 \times 10^{-4} & \\ 0.46 \times 10^{-3} & 0.12 \times 10^{-3} & -0.42 \times 10^{-4} & 0.41 \times 10^{-3} \end{pmatrix} \quad [3.3.8]$$

Por tanto, estas estimaciones no difieren significativamente de las obtenidas por *MVE*, aunque puede apreciarse un ligero aumento en las varianzas estimadas de los residuos respecto a las obtenidas con los algoritmos **[MAXFV]** y **[FVE]** (*vid.* [3.1.35]-[3.1.36]). También es interesante señalar que el parámetro α estimado es mayor que la unidad, lo cual es imposible que ocurra cuando se está maximizando la *FVE* (*vid.* apartados 2.2.4, 2.3.2 y 3.2.2).

Las pequeñas diferencias entre los resultados obtenidos con los algoritmos **[FVE]** y **[MAXFV]** y los obtenidos con *The S.C.A. Statistical System* podrían sugerir que, en la práctica, ambos procedimientos son equivalentes. Sin embargo, no resulta difícil encontrar ejemplos reales que rechazan claramente esta posible equivalencia. Para ilustrar el mejor comportamiento de los algoritmos **[FVE]** y **[MAXFV]** respecto a *The S.C.A. Statistical System*, se ha ajustado un modelo $ARIMA(0,1,1) \times (2,1,1)_{12}$ univariante a dos series económicas mensuales (Consumo Aparente de Cemento e I.P.I. Energía) de tamaño $n = 107$. Los resultados obtenidos con los dos procedimientos citados (que se presentan en las hojas resumen 1 a 8) son claramente distintos cuando hay anomalías importantes dentro de las 24 (orden AR) primeras observaciones de la muestra. Esto es debido a que dichas observaciones son tratadas por *The S.C.A. Statistical System* como números fijos para calcular una serie que se supone generada por la parte MA del proceso (*vid.* apartado 2.2.4). Más importante aún es observar que estas situaciones no pueden detectarse utilizando *The S.C.A. Statistical System*, ya que los 24 primeros residuos no son calculados.

Una comparación entre las hojas resumen 1 y 2, es suficiente para poner claramente de manifiesto lo inadecuado del procedimiento de estimación empleado por *The S.C.A. Statistical System* cuando la muestra considerada contiene valores anómalos dentro de las p primeras observaciones. La diferencia más interesante entre ambas hojas se encuentra en las estimaciones del parámetro MA estacional, que resulta no invertible cuando el modelo es estimado con los algoritmos [MAXFV] y [FVE], mientras que es claramente distinto de 1 (incluso prácticamente no significativo) cuando el modelo es estimado con *The S.C.A. Statistical System*. También es interesante señalar que el posible carácter influyente de las observaciones 2/83, 3/84 y 1/85 no podría detectarse mediante la inspección del gráfico de residuos de la hoja resumen 2, ya que los residuos correspondientes al período 2/83-1/85 (24 observaciones) no son calculados por *The S.C.A. Statistical System*. Por último, es posible señalar que el funcionamiento de los algoritmos [MAXFV] y [FVE] sí puede considerarse adecuado en este tipo de situaciones, ya que las estimaciones resumidas en las hojas 3 y 4 (con las observaciones 2/83, 3/84 y 1/85 corregidas) no difieren sensiblemente de las presentadas en la hoja 1; concretamente, en los tres casos el parámetro MA estacional resulta no invertible (esta situación podría sugerir que la estacionalidad de la serie no puede captarse adecuadamente mediante la modelización utilizada, pero este es un tema que cae fuera del ámbito del presente trabajo).

Una situación similar es la que se presenta en las hojas resumen 5-8. En este caso, el parámetro MA estacional sí es claramente invertible (como se muestra en las hojas 5, 7 y 8), aunque cuando el modelo es estimado con *The S.C.A. Statistical System* sin corregir el dato 1/85 (cuyo residuo no aparece en el gráfico de la hoja 6), dicho parámetro resulta no invertible. En definitiva, las estimaciones generadas por el procedimiento de Hillmer y Tiao (1979) pueden resultar inadecuadas en ciertos casos, casos en los que, por otro lado, los algoritmos [MAXFV] y [FVE] sí funcionan correctamente.

Para concluir este capítulo, puede señalarse que, debido a la coincidencia en los valores numéricos de la *FVE* evaluada con los algoritmos de Shea (1984, 1987, 1989) y [FVE] (*vid.* Sección 2.4), no cabe esperar diferencias apreciables entre las estimaciones por *MVE* obtenidas con el algoritmo [MAXFV] y con la subrutina G13DCF de la librería *N.A.G.* (1987), basada en Shea (1987). En concreto, si el algoritmo de optimización empleado en ambos procedimientos fuese el mismo (*vid.* Shea (1984), pp. 99-100), los resultados deberían ser muy similares. No obstante, el algoritmo [FVE] presenta una serie de ventajas prácticas (*vid.* secciones 2.3 y 2.4) que sugieren su utilización en la mayoría de las situaciones que ocurren en la práctica econométrica habitual.

CAPITULO 4

Conclusiones

La estimación eficiente de un modelo ARMA constituye una fase crítica dentro del proceso iterativo de modelización de series temporales, ya que de sus resultados (incluyendo los instrumentos que proporciona para la fase de diagnóstico) dependen en gran medida las decisiones que se toman sobre la posible reformulación del modelo considerado.

En situaciones bien condicionadas, caracterizadas por muestras largas, modelos claramente estacionarios e invertibles y ausencia de anomalías importantes, no suele ser necesario recurrir a la estimación por *MVE* para obtener resultados satisfactorios. En dichas situaciones, otros métodos de estimación más sencillos, computacionalmente más eficientes y con las mismas propiedades asintóticas (*MVC*, *MSC con retrovisión* o *MSC condicional*) pueden proporcionar resultados que no difieren significativamente de los obtenidos por *MVE*.

No obstante, en la práctica, no es difícil encontrar situaciones en las que los resultados obtenidos con un método difieren sustancialmente de los obtenidos con otro. En concreto, en situaciones caracterizadas por muestras cortas, modelos cercanos a la no invertibilidad o a la no estacionariedad o por la presencia de anomalías importantes en la muestra, es muy recomendable utilizar el criterio de estimación por *MVE*. De lo contrario, los resultados obtenidos pueden conducir a conclusiones erróneas sobre lo adecuado o la posible reformulación de un modelo. Por supuesto, la estimación por *MVE* también funciona correctamente en situaciones bien condicionadas, por lo que su empleo debería considerarse más como una norma que como una posible alternativa.

Teniendo en cuenta este tipo de conclusiones (que no son nuevas), en este trabajo se ha diseñado un nuevo procedimiento para la estimación por *MVE* de procesos ARMA multivariantes estocásticos. Las principales aportaciones respecto a la literatura existente sobre la materia, pueden resumirse en los siguientes puntos:

- [1] Se ha diferenciado claramente entre *evaluación* (Capítulo 2) y *maximización* (Capítulo 3) de la *FVE*. Esta distinción ha permitido estudiar soluciones detalladas a ambos problemas, en lugar de proponer

la aplicación rutinaria de un algoritmo estándar de optimización numérica a una formulación más o menos adecuada de la *FVE*.

- [2] Se han integrado las soluciones propuestas a ambos problemas en un algoritmo completo de estimación por *MVE*, cuyo funcionamiento en la práctica ha sido contrastado (Capítulo 3) mediante un amplio conjunto de estimaciones en situaciones con diferentes estructuras de condicionamiento, tanto simuladas como reales.
- [3] Dado que la utilidad de un trabajo de estas características debe contrastarse, sobre todo, en la práctica, se ha prestado especial atención a facilitar la utilización efectiva de los algoritmos propuestos, de manera que, junto con un análisis teórico riguroso, se han descrito (Apéndices A.2 y A.3) todos los detalles necesarios para su codificación en algún lenguaje de programación.

Además de estas aportaciones de carácter general, respecto al nuevo algoritmo propuesto para evaluar la *FVE* (algoritmo **[FVE]**) pueden destacarse las siguientes:

- [1] Se han explotado al máximo las posibilidades ofrecidas y no exploradas en los trabajos más relevantes sobre el tema.
- [2] A consecuencia de lo anterior, se han reunido, en un sólo algoritmo, todas las ventajas que se encuentran por separado en los procedimientos disponibles en la actualidad, a la vez que se han evitado sus posibles inconvenientes.

En concreto, el algoritmo **[FVE]** (Capítulo 2) evalúa la función de verosimilitud de forma *analíticamente exacta* [algo no contemplado, por ejemplo, en Hillmer y Tiao (1979)] y permite realizar una evaluación *aproximada*, que es numéricamente precisa y considerablemente más rápida que la evaluación exacta [propiedad compartida con el algoritmo de Shea (1989)]. Por otro lado, el algoritmo **[FVE]** permite detectar *como subproducto de ciertos cálculos* situaciones de no estacionariedad [imposibles de identificar con el algoritmo de Hillmer y Tiao (1979)] y de no invertibilidad [imposibles de identificar con el algoritmo de Shea (1989)]. Por último, el algoritmo **[FVE]** proporciona los residuos *analíticamente exactos* correspondientes a la especificación considerada [no calculables directamente con los algoritmos de Hillmer y Tiao (1979) y de Shea (1989)] y resulta superior, en términos de eficiencia computacional, a los algoritmos de Hall y Nicholls (1979) y a la extensión multivariante de Ljung y Box (1979), a la vez que puede competir en estos términos (y resultar superior en muchos casos) con el algoritmo de Shea (1989).

Con el fin de aprovechar las ventajas ofrecidas por el algoritmo [FVE], se ha diseñado un procedimiento de maximización de la *FVE* (algoritmo [MAXFV]), basado en uno de los métodos de optimización numérica más eficientes disponibles en la actualidad (un método *cuasi-Newton*, basado en la versión factorizada de la actualización BFGS, que utiliza primeras derivadas evaluadas mediante diferencias finitas). La simple descripción de los detalles para acoplar dicho método a las peculiaridades de la función objetivo (derivada a partir de la *FVE*), constituye una interesante aportación desde el punto de vista práctico, ya que esta cuestión ha sido pasada por alto sistemáticamente en los trabajos existentes sobre estimación de modelos ARMA [por ejemplo, en Hillmer y Tiao (1979) y Hall y Nicholls (1980)].

El resultado de la combinación de los algoritmos [FVE] y [MAXFV] es un procedimiento de estimación computacionalmente eficiente que, además de proporcionar estimaciones adecuadas y los medios necesarios para llevar a cabo la diagnosis del modelo estimado, permite demostrar cómo algunos procedimientos, teóricamente válidos, pueden fallar en la práctica (como se ilustra, respecto al algoritmo de Hillmer y Tiao (1979), en los últimos ejemplos del Capítulo 3).

Aparte de algunas extensiones inmediatas (como la incorporación del algoritmo de Kohn y Ansley (1982) para el cálculo de las autocovarianzas teóricas), a lo largo del proceso de elaboración de este trabajo, han surgido varias cuestiones cuya solución se puede ver facilitada por los algoritmos desarrollados en los Capítulos 2 y 3. En concreto, creemos que sería interesante investigar los siguientes temas:

- [1] La estimación simultánea de un modelo ARMA junto con los componentes deterministas asociados a un vector de series temporales.
- [2] La estimación simultánea de la estructura ARMA junto con la matriz de coeficientes de *cointegración*, en modelos multivariantes con series *no estacionarias cointegradas*.
- [3] La estimación de representaciones alternativas a la estacionalidad estocástica multiplicativa, un ejemplo de cuya posible presencia se encuentra al final del Capítulo 3.
- [4] El diseño de nuevos instrumentos de diagnosis de la estimación, basados en el análisis y en la integración numérica de curvas semejantes a las presentadas en el Capítulo 3.

El análisis de estos temas, sobre la base de los resultados obtenidos en este trabajo, puede resultar muy útil a la hora de mejorar las técnicas disponibles para la modelización de series temporales.

APENDICES

APENDICE A.1

Resultados de álgebra matricial

En este apéndice se presentan los instrumentos de álgebra matricial utilizados en los Capítulos 2 y 3 y en el Apéndice A.4. A continuación de cada resultado se cita una referencia donde puede encontrarse su demostración, aunque en general, dicha demostración también puede encontrarse, en diferentes versiones, en muchos de los manuales de Econometría más conocidos. Tan sólo en dos casos se presenta la demostración a continuación del enunciado, ya que tales resultados no figuran en ninguna de las referencias consultadas.

[A.1.1] Sean A y B dos matrices de orden $(m \times n)$ y $(p \times q)$, respectivamente. El *producto de Kronecker* de las dos matrices se denota por $A \otimes B$ y se define como la siguiente matriz de orden $(mp \times nq)$:

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{pmatrix}$$

(*vid.* Dhrymes (1984), pp. 449-52).

[A.1.2] Sea A una matriz $(m \times n)$. Se representa por $vec(A)$ al vector columna de mn elementos formado por las n columnas de A dispuestas una debajo de la otra. Es decir, si denotamos la i -ésima columna de A por a_i ($i = 1, 2, \dots, n$), entonces $vec(A) = [a_1^T, a_2^T, \dots, a_n^T]^T$ (*vid.* Dhrymes (1984), pp. 508-13).

[A.1.3] Sean A y B dos matrices de orden $(m \times n)$ y $(n \times q)$, respectivamente. Entonces, puede demostrarse que $vec(AB) = (B^T \otimes I_m)vec(A) = (I_q \otimes A)vec(B)$ (*vid.* Dhrymes (1984), pp. 509).

[A.1.4] Si A es una matriz cuadrada de orden n , entonces existe una matriz $(n^2 \times n^2)$ de *permutación* P tal que $Pvec(A) = vec(A^T)$ (*vid.* Nicholls y Hall (1979), pp. 260). En concreto, todos los elementos de la matriz P son cero, excepto $P_{i+kn, k+1+(i-1)n} = 1$ ($i = 1, 2, \dots, n; k = 0, 1, \dots, n-1$).

[A.1.5] Sean A , P y Q matrices de orden $(n \times n)$, $(p \times p)$ y $(n \times p)$, respectivamente, con A no singular. Entonces $(A + PQP^T)^{-1} = A^{-1} - A^{-1}Q[P^{-1} + Q^T A^{-1}Q]^{-1}Q^T A^{-1}$, siempre que $(P^{-1} + Q^T A^{-1}Q)$ sea no singular (vid. Maddala, pp. 445-47).

[A.1.6] Sea A una matriz cuadrada no singular, cuyos elementos son funciones de un escalar α . Entonces:

$$\frac{\partial |A|}{\partial \alpha} = |A| \times \text{traza} \left[A^{-1} \frac{\partial A}{\partial \alpha} \right]$$

(vid. Dhrymes (1984), pp. 523-5).

[A.1.7] Sea A una matriz cuadrada no singular, cuyos elementos son funciones de un escalar α . Entonces:

$$\frac{\partial A^{-1}}{\partial \alpha} = -A^{-1} \frac{\partial A}{\partial \alpha} A^{-1}$$

(vid. Dhrymes (1984), pp. 529-30).

[A.1.8] Sea Q una matriz simétrica y definida positiva tal que $Q^{-1} = R^T R$ y cuyos elementos son funciones de un escalar α . Entonces:

$$\frac{\partial R}{\partial \alpha} = -\frac{1}{2} R \frac{\partial Q}{\partial \alpha} Q^{-1}$$

Demostración. En primer lugar, teniendo en cuenta [A.1.7] y que $Q^{-1} = R^T R$:

$$\begin{aligned} \frac{\partial Q^{-1}}{\partial \alpha} &= -Q^{-1} \frac{\partial Q}{\partial \alpha} Q^{-1} \\ &= -\frac{1}{2} Q^{-1} \frac{\partial Q}{\partial \alpha} Q^{-1} - \frac{1}{2} Q^{-1} \frac{\partial Q}{\partial \alpha} Q^{-1} \\ &= -\frac{1}{2} Q^{-1} \frac{\partial Q}{\partial \alpha} R^T R - \frac{1}{2} R^T R \frac{\partial Q}{\partial \alpha} Q^{-1} \end{aligned}$$

Y en segundo lugar:

$$\frac{\partial (R^T R)}{\partial \alpha} = \frac{\partial R^T}{\partial \alpha} R + R^T \frac{\partial R}{\partial \alpha}$$

Como $Q^{-1} = R^T R$, estas dos expresiones deben coincidir, por lo que:

$$-\frac{1}{2} \mathbf{Q}^{-1} \frac{\partial \mathbf{Q}}{\partial \alpha} \mathbf{R}^T \mathbf{R} - \frac{1}{2} \mathbf{R}^T \mathbf{R} \frac{\partial \mathbf{Q}}{\partial \alpha} \mathbf{Q}^{-1} = \frac{\partial \mathbf{R}^T}{\partial \alpha} \mathbf{R} + \mathbf{R}^T \frac{\partial \mathbf{R}}{\partial \alpha}$$

de donde se deduce inmediatamente el resultado propuesto \square .

[A.1.9] Sea \mathbf{M} una matriz simétrica y definida positiva tal que $\mathbf{M} = \mathbf{L}\mathbf{L}^T$ y cuyos elementos son funciones de un escalar α . Entonces:

$$\frac{\partial \mathbf{L}}{\partial \alpha} = -\frac{1}{2} \mathbf{M} \frac{\partial \mathbf{M}^{-1}}{\partial \alpha} \mathbf{L}$$

Demostración. Si definimos $\mathbf{W}^{-1} = \mathbf{M} = \mathbf{L}\mathbf{L}^T$, teniendo en cuenta [A.1.7]:

$$\begin{aligned} \frac{\partial \mathbf{W}^{-1}}{\partial \alpha} &= -\mathbf{W}^{-1} \frac{\partial \mathbf{W}}{\partial \alpha} \mathbf{W}^{-1} \\ &= -\frac{1}{2} \mathbf{W}^{-1} \frac{\partial \mathbf{W}}{\partial \alpha} \mathbf{W}^{-1} - \frac{1}{2} \mathbf{W}^{-1} \frac{\partial \mathbf{W}}{\partial \alpha} \mathbf{W}^{-1} \\ &= -\frac{1}{2} \mathbf{W}^{-1} \frac{\partial \mathbf{W}}{\partial \alpha} \mathbf{L}\mathbf{L}^T - \frac{1}{2} \mathbf{L}\mathbf{L}^T \frac{\partial \mathbf{W}}{\partial \alpha} \mathbf{W}^{-1} \end{aligned}$$

Por otro lado:

$$\frac{\partial(\mathbf{L}\mathbf{L}^T)}{\partial \alpha} = \frac{\partial \mathbf{L}}{\partial \alpha} \mathbf{L}^T + \mathbf{L} \frac{\partial \mathbf{L}^T}{\partial \alpha}$$

Como $\mathbf{W}^{-1} = \mathbf{L}\mathbf{L}^T$, estas dos expresiones deben coincidir, por lo que:

$$-\frac{1}{2} \mathbf{W}^{-1} \frac{\partial \mathbf{W}}{\partial \alpha} \mathbf{L}\mathbf{L}^T - \frac{1}{2} \mathbf{L}\mathbf{L}^T \frac{\partial \mathbf{W}}{\partial \alpha} \mathbf{W}^{-1} = \frac{\partial \mathbf{L}}{\partial \alpha} \mathbf{L}^T + \mathbf{L} \frac{\partial \mathbf{L}^T}{\partial \alpha}$$

de donde:

$$\frac{\partial \mathbf{L}}{\partial \alpha} = -\frac{1}{2} \mathbf{W}^{-1} \frac{\partial \mathbf{W}}{\partial \alpha} \mathbf{L}$$

que, teniendo en cuenta que $\mathbf{W}^{-1} = \mathbf{M}$, es el resultado propuesto \square .

APENDICE A.2

Codificación del algoritmo [FVE]

Introducción

En este apéndice y en el siguiente se presentan, en forma de *pseudocódigo*, los algoritmos [FVE] y [MAXFV] desarrollados en los Capítulos 2 y 3. El objetivo fundamental consiste en proporcionar una descripción lo suficientemente detallada de estos algoritmos como para que su codificación en algún lenguaje de alto nivel (*Pascal*, *FORTRAN* o *C*) resulte razonablemente sencilla. Con ello, se pretende ofrecer todo lo necesario para disponer en la práctica de un método de estimación preciso y computacionalmente eficiente, que puede utilizarse por sí sólo o, mejor aún, en combinación con otras herramientas propias del análisis de series temporales multivariantes (*vid.* Capítulo 4).

Dado su carácter, la lectura de estos apéndices puede resultar incómoda (incluso ininteligible) para quienes no estén familiarizados con la programación de ordenadores. No obstante, se ha considerado fundamental su inclusión en este trabajo, ya que la exposición de los algoritmos [FVE] y [MAXFV] en un formato similar y fácilmente convertible a un lenguaje de programación de alto nivel, permite:

- [1] Proporcionar un grado de detalle en la descripción de los algoritmos imposible de alcanzar sólo con texto y fórmulas. De hecho, muchos de los aspectos prácticos simplemente apuntados en los Capítulos 2 y 3, se desarrollan completamente en estos apéndices, permitiendo así una mayor comprensión del funcionamiento de los algoritmos en la práctica.
- [2] Mostrar que los algoritmos propuestos en el texto pueden ponerse a funcionar en la práctica sin grandes dificultades. De esta forma, se pretende subrayar la importancia de disponer de técnicas que, basadas en argumentos teóricos rigurosos, sean útiles y eficaces en la práctica habitual.
- [3] Proporcionar, en fin, todos los medios necesarios (junto con el Apéndice A.5) para contrastar la validez de los resultados empíricos presentados en los Capítulos 2 y 3, así como para comparar dichos resultados (o cualesquiera otros relativos a distintas especificaciones) con los obtenidos mediante otras técnicas maximoverosímiles o incluso mediante otros criterios de estimación.

El *pseudocódigo* se ha elaborado a partir del código fuente en *Pascal*, siguiendo las pautas descritas en Dennis y Schnabel (1983), pp. 262-71, que no se transcriben aquí. No obstante, como complemento a dichas pautas, se describen a continuación las reglas que se han seguido en estos apéndices para almacenar ciertas cantidades que entran en juego en los algoritmos [FVE] y [MAXFV], así como las alternativas más relevantes a dichas reglas.

El almacenamiento de los elementos de un vector $(n \times 1)$ $v \in E^n$ y de una matriz $(n \times n)$ $M \in E^{n \times n}$, se ha llevado a cabo asignando, respectivamente, el i -ésimo elemento de v a $vec[i]$ ($i = 1, 2, \dots, n$) y el elemento (i, j) -ésimo de M a $MAT[i, j]$, donde vec y MAT son dos variables de tipo "array" definidas de acuerdo con el lenguaje de programación empleado (p.e. " $vec: ARRAY[1..n]$ OF real" en *Pascal* o " $double MAT[n][n]$ " en C). Para almacenar los elementos de una secuencia de matrices $M_k \in E^{n \times n}$ ($k = 1, 2, \dots, l$), se ha asignado el elemento (i, j) -ésimo de M_k a $MAT(k)[i, j]$ ($k = 1, 2, \dots, l; i, j = 1, 2, \dots, n$). Aunque esta es la forma más intuitiva de almacenar los elementos de $M_k \in E^{n \times n}$ ($k = 1, 2, \dots, l$), su instrumentación tan sólo es inmediata en *Pascal* (p.e. " $TYPE matnn: ARRAY[1..n, 1..n]$ OF real; VAR MAT: ARRAY[1..l] OF matnn"). En *FORTRAN* y C, resulta conveniente asignar el elemento (i, j) -ésimo de M_k a $MAT[i, j + (k-1)n]$ ($k = 1, 2, \dots, l; i, j = 1, 2, \dots, n$), donde MAT es ahora una variable de tipo "array" (p.e. definida en C como " $double MAT[n][ln]$ "). Esta ha sido, de hecho, la regla seguida para almacenar los elementos de una secuencia de vectores $v_k \in E^n$ ($k = 1, 2, \dots, l$), de forma que se ha asignado el elemento i -ésimo de v_k a $vec[i + (k-1)n]$ ($k = 1, 2, \dots, l; i = 1, 2, \dots, n$), donde vec es una variable de tipo "array" (p.e. " $vec: ARRAY[1..ln]$ OF real" en *Pascal* o " $double vec[ln]$ " en C). Por último, el problema de almacenar los elementos de una secuencia de matrices $M_k \in E^{n \times n}$ ($k = 0, 1, \dots, l$), se ha resuelto asignando el elemento (i, j) -ésimo de M_k a $MAT(k)[i, j]$ en algún caso y a $vec[kn^2 + (i-1)n + j]$ en otros ($k = 0, 1, \dots, l; i, j = 1, 2, \dots, n$), contemplándose así tanto las facilidades ofrecidas por *Pascal* como los condicionantes de *FORTRAN* y C (vid. algoritmo CGAMMA).

Los algoritmos que se describen en este apéndice están englobados en tres categorías: algoritmo principal (FVE), algoritmos intermedios (CGAMMA, CXI y CRES) y algoritmos auxiliares (LUDCMP, LUSOL, CHOLDCMP, CHOLBACK y CHOLFOR). El algoritmo principal tiene el propósito de evaluar la FVE (o la FVA) del modelo especificado. Los cálculos de las matrices de covarianzas teóricas, de la secuencia de matrices [2.1.24] y del vector de residuos (opcional) se llevan a cabo en los algoritmos intermedios. Finalmente, tanto el algoritmo principal como los intermedios hacen uso de los algoritmos auxiliares, que contienen procedimientos de cálculo matricial estándar (excepto el algoritmo CHOLDCMP), utilizados también por el algoritmo MAXFVCN (vid. Apéndice A.3). Por tanto, para incorporar todas estas rutinas en un sistema de análisis multivariante más amplio (vid. Apéndice A.3),

tan sólo se requiere una llamada al algoritmo principal. La Figura A.3.1 muestra las dependencias y llamadas entre el algoritmo **FVE** (principal) y los algoritmos intermedios y auxiliares.

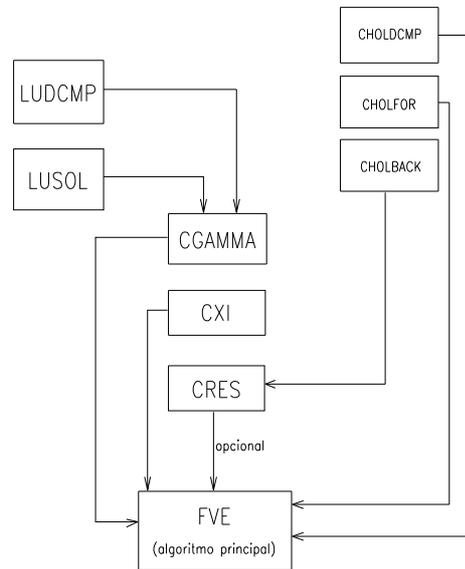


Fig. A.3.1: Diagrama de dependencias del algoritmo **FVE**.

Con el fin de centrar la atención sobre los métodos propios de la evaluación de la *FVE* de modelos ARMA multivariantes, en el apartado **Descripción** de algunos algoritmos auxiliares puede encontrarse, en lugar de una descripción del algoritmo, una referencia donde dicha descripción se encuentra con todo detalle (también para evitar en lo posible la transcripción literal de ciertas fuentes). Por último, puede señalarse que, en general, se ha intentado que el *pseudocódigo* resulte lo más eficiente posible en términos de necesidades de almacenamiento. No obstante, cuando se ha considerado algún artificio para ahorrar memoria demasiado complicado u oscuro, se ha sacrificado eficiencia por claridad (naturalmente, esta no ha sido la regla a la hora de escribir el código fuente). Salvo por estas excepciones, la redacción del *pseudocódigo* se ha efectuado directamente sobre el código fuente en *Pascal*, por lo que esperamos que la posibilidad de encontrar algún error en el mismo haya quedado reducida al mínimo.

Algoritmo LUDCMP

Propósito.- Descomposición de una matriz cuadrada $M' = LU$. M' está formada mediante permutaciones de las filas de la matriz M (que puede ser una matriz a invertir o la matriz de un sistema de ecuaciones a resolver), L es triangular inferior con unos en la diagonal principal y U es triangular

superior (*vid.* Press, Flannery, Teukolsky y Vetterling (1989), cap. 2).

Llamada.- LUDCMP(*MAT*, *n*, *indx*, *dcmpok*).

Parámetros de entrada.- $n \in \mathbb{Z}$: dimensión de la matriz a descomponer.

Parámetros de entrada-salida.- $MAT \in E^{n \times n}$: a la entrada, el elemento (*i,j*)-ésimo de *M* (la matriz original, no la permutada) se almacena en *MAT*[*i,j*] (*i,j* = 1,2,...,*n*); a la salida, *MAT*[*i,j*] (*i* = 2,3,...,*n*; *j* = 1,2,...,*i*-1; triángulo inferior de *MAT*) contiene el elemento (*i,j*)-ésimo de *L* y *MAT*[*i,j*] (*i* = 1,2,...,*n*; *j* = *i,i*+1,...,*n*; diagonal principal y triángulo superior de *MAT*) contiene el elemento (*i,j*)-ésimo de *U* (la descomposición de la matriz permutada *M'*).

Parámetros de salida.- [1] *indx* $\in \mathbb{Z}^n$: vector de números enteros que registra las permutaciones efectuadas sobre la matriz original. [2] *dcmpok* $\in \{\text{TRUE}, \text{FALSE}\}$: cuando la matriz *M* es singular, el algoritmo devuelve *dcmpok* = FALSE; en caso contrario, devuelve *dcmpok* = TRUE.

Almacenamiento intermedio.- Un vector auxiliar *vtmp* $\in E^n$.

Otras consideraciones.- Este algoritmo se utiliza junto con LUSOL para resolver el sistema de ecuaciones lineales [2.2.12].

Descripción.- El funcionamiento y la descripción detallada de este algoritmo pueden encontrarse en Press, Flannery, Teukolsky y Vetterling (1989), pp. 39-47.

Algoritmo:

(* Asignación de valores iniciales *)

1: *dcmpok* \leftarrow TRUE.

(* Cálculo de la mayor componente en cada fila *)

2: FOR *i* = 1 TO *n* DO

2.1: *mfil* \leftarrow $\max_{1 \leq j \leq n} (|MAT[i,j]|)$.

2.2: IF (*mfil* = 0.0) THEN

(* Matriz singular \Rightarrow fin anormal algoritmo LUDCMP *)

2.2.1: *dcmpok* \leftarrow FALSE.

2.2.2: RETURN.

2.3: *vtmp*[*i*] \leftarrow 1.0 / *mfil*.

(* Bucle principal: algoritmo de Crout con pivoteo parcial *)

3: FOR *j* = 1 TO *n* DO

3.1: FOR *i* = 1 TO *j*-1 DO

$$MAT[i,j] \leftarrow MAT[i,j] - \sum_{k=1}^{i-1} MAT[i,k] \times MAT[k,j].$$

3.2: *mfil* \leftarrow 0.0.

3.3: FOR *i* = *j* TO *n* DO

3.3.1: $MAT[i,j] \leftarrow MAT[i,j] - \sum_{k=1}^{j-1} MAT[i,k] \times MAT[k,j].$

3.3.2: $tmp \leftarrow vtmp[i] \times |MAT[i,j]|.$

```

3.3.3: IF ( $tmp \geq mfil$ ) THEN
    3.3.3.1:  $mfil \leftarrow tmp$ .
    3.3.3.2:  $imax \leftarrow i$ .
(* END FOR 3.3 *)
3.4: IF ( $j \neq imax$ ) THEN
    3.4.1: FOR  $k = 1$  TO  $n$  DO
        3.4.1.1:  $tmp \leftarrow MAT[imax,k]$ .
        3.4.1.2:  $MAT[imax,k] \leftarrow MAT[j,k]$ .
        3.4.1.3:  $MAT[j,k] \leftarrow tmp$ .
    3.4.2:  $vtmp[imax] \leftarrow vtmp[j]$ .
(* END IF 3.4 *)
3.5:  $indx[j] \leftarrow imax$ .
3.6: IF ( $MAT[j,j] = 0.0$ ) THEN
     $MAT[j,j] \leftarrow 1.0^{-20}$ .
3.7: IF ( $j \neq n$ ) THEN
    3.7.1:  $tmp \leftarrow 1.0 / MAT[j,j]$ .
    3.7.2: FOR  $i = j+1$  TO  $n$  DO
         $MAT[i,j] \leftarrow MAT[i,j] \times tmp$ .
(* END IF 3.7 *)
(* END FOR 3 *)
4: RETURN. (* Fin normal algoritmo LUDCMP *)

```

Algoritmo LUSOL

Propósito.- Solución del sistema de n ecuaciones lineales $Mx = b$, haciendo uso de la descomposición de la matriz M calculada por el algoritmo LUDCMP.

Llamada.- LUSOL($MAT, n, indx, rhsol$).

Parámetros de entrada.- [1] $MAT \in E^{n \times n}$: descomposición de la matriz M del sistema, según es devuelta por el algoritmo LUDCMP. [2] $n \in Z$: dimensión del sistema (n^0 de ecuaciones). [3] $indx \in Z^n$: vector de números enteros devuelto por el algoritmo LUDCMP.

Parámetros de entrada-salida.- $rhsol \in E^n$: a la entrada, el elemento i -ésimo de b (el lado derecho del sistema) se almacena en $rhsol[i]$ ($i = 1, 2, \dots, n$); a la salida, $rhsol[i]$ ($i = 2, 3, \dots, n$) contiene el elemento i -ésimo de x (la solución del sistema).

Otras consideraciones.- Este algoritmo se utiliza junto con LUDCMP para resolver el sistema de ecuaciones lineales [2.2.12] (vid. algoritmo CGAMMA).

Descripción.- El funcionamiento y la descripción detallada de este algoritmo pueden encontrarse en

Press, Flannery, Teukolsky y Vetterling (1989), pp. 39-47.

Algoritmo:

```

1:  $k \leftarrow 0$ .
(* "Forward substitution" *)
2: FOR  $i = 1$  TO  $n$  DO
    2.1:  $ix \leftarrow \text{indx}[i]$ .
    2.2:  $\text{tmp} \leftarrow \text{rhsol}[ix]$ .
    2.3:  $\text{rhsol}[ix] \leftarrow \text{rhsol}[i]$ .
    2.4T: IF ( $k \neq 0$ ) THEN
         $\text{tmp} \leftarrow \text{tmp} - \sum_{j=k}^{i-1} \text{MAT}[i,j] \times \text{rhsol}[j]$ .
    2.4E: ELSE IF ( $\text{tmp} \neq 0.0$ ) THEN
         $k \leftarrow i$ .
    2.5:  $\text{rhsol}[i] \leftarrow \text{tmp}$ .
(* END FOR 2 *)
(* "Backsubstitution" *)
3: FOR  $i = n$  DOWNTO 1 DO
    3.1:  $\text{tmp} \leftarrow \text{rhsol}[i] - \sum_{j=i+1}^n \text{MAT}[i,j] \times \text{rhsol}[j]$ .
    3.2:  $\text{rhsol}[i] \leftarrow \text{tmp} / \text{MAT}[i,i]$ .
4: RETURN. (* Fin normal algoritmo LUSOL *)

```

Algoritmo CHOLDCMP

Propósito.- [1] Encontrar la descomposición LL^T (Cholesky) de una matriz M simétrica y definida positiva, perturbada según el algoritmo de Gill, Murray y Wright (1981) cuando se detecta que dicha matriz *no* es definida positiva por errores de redondeo. [2] Cálculo del determinante de M .

Llamada.- CHOLDCMP(MAT, n, d1, d2, dpok, epsmaq).

Parámetros de entrada.- [1] $n \in \mathbb{Z}$: dimensión de la matriz a descomponer. [2] $\text{epsmaq} \in E$: épsilon máquina (vid. algoritmo CEPSMQ en el Apéndice A.3).

Parámetros de entrada-salida.- $\text{MAT} \in E^{n \times n}$: a la entrada, $\text{MAT}[i,j]$ contiene el elemento (i,j) -ésimo de M (tan sólo es necesario el triángulo superior y la diagonal principal); a la salida, $\text{MAT}[i,j]$ contiene el elemento (i,j) -ésimo de L ($i = 1, 2, \dots, n; j = 1, 2, \dots, i$; nótese que L es triangular inferior).

Parámetros de salida.- [1] $d1 \in E$: factor a en la expresión del determinante $|M| = a2^b$. [2] $d2 \in \mathbb{Z}$: factor b en la expresión del determinante $|M| = a2^b$. [3] $\text{dpok} \in \{\text{TRUE}, \text{FALSE}\}$: el algoritmo devuelve $\text{dpok} = \text{FALSE}$ cuando M *no* es definida positiva y $\text{dpok} = \text{TRUE}$ cuando sí lo es.

Descripción (*vid.* Dennis y Schnabel (1983), pp. 318-9).- El algoritmo intenta obtener la descomposición de Cholesky estándar de M , a la vez que va actualizando los factores a y b para el cálculo de su determinante. No obstante, cuando se detecta que M no es definida positiva debido a errores numéricos, se perturban los elementos de su diagonal principal lo justo para garantizar su buen condicionamiento; si el carácter no definido positivo no es debido a errores numéricos, el algoritmo se interrumpe devolviendo $dpok = \text{FALSE}$.

Algoritmo:

(* Asignación de valores iniciales *)

1: $dpok \leftarrow \text{TRUE}$.

2: $d1 \leftarrow 1.0$.

3: $d2 \leftarrow 0$.

4: $minl \leftarrow 0.0$.

5: $maxoffl \leftarrow \sqrt{\max_{1 \leq i \leq n} (|MAT[i,i]|)}$.

6: $minl2 \leftarrow (epsmaq)^{1/2} \times maxoffl$.

7: $maxadd \leftarrow 0.0$.

(* Bucle principal: cálculo de la j -ésima columna de L *)

8: FOR $j = 1$ TO n DO

8.1: $MAT[j,j] \leftarrow MAT[j,j] - \sum_{i=1}^{j-1} (MAT[j,i])^2$.

8.2: IF $(MAT[j,j] \neq |MAT[j,j]|)$ AND $(|MAT[j,j]| > minl2)$ THEN

(* Matriz M no definida positiva \Rightarrow fin anormal algoritmo **CHOLDCMP** *)

8.2.1: $dpok \leftarrow \text{FALSE}$.

8.2.2: RETURN.

8.3: $minljj \leftarrow 0.0$.

8.4: FOR $i = j+1$ TO n DO

8.4.1: $MAT[i,j] \leftarrow MAT[j,i] - \sum_{k=1}^{j-1} MAT[i,k] \times MAT[j,k]$.

8.4.2: $minljj \leftarrow \max(|MAT[i,j]|, minljj)$.

8.5: $minljj \leftarrow \max(minljj / maxoffl, minl)$.

8.6: IF $(MAT[j,j] > minljj^2)$

8.6T: THEN (* Descomposición de Cholesky estándar *)

$MAT[j,j] \leftarrow (MAT[j,j])^{1/2}$.

8.6E: ELSE (* Perturbación de la diagonal principal *)

8.6E.1: IF $(minljj < minl2)$ THEN

$minljj \leftarrow minl2$.

8.6E.2: $maxadd \leftarrow \max(maxadd, minljj^2 - MAT[j,j])$.

8.6E.3: $MAT[j,j] \leftarrow minljj$.

```
(* Actualización de los factores para el cálculo de  $|M|$  *)
8.7:  $d1 \leftarrow d1 \times (MAT[j,j])^2$ .
8.8: WHILE ( $|d1| \geq 1.0$ ) DO
    8.8.1:  $d1 \leftarrow d1 \times 0.0625$ .
    8.8.2:  $d2 \leftarrow d2 + 4$ .
8.9: WHILE ( $|d1| < 0.0625$ ) DO
    8.9.1:  $d1 \leftarrow d1 \times 16.0$ .
    8.9.2:  $d2 \leftarrow d2 - 4$ .
8.10: FOR  $i = j+1$  TO  $N$  DO
     $MAT[i,j] \leftarrow MAT[i,j] / MAT[j,j]$ .
(* END FOR 8 *)
9: RETURN. (* Fin normal algoritmo CHOLDCMP *)
```

Algoritmo CHOLFOR

Propósito.- Solución del sistema de n ecuaciones lineales $Lx = b$, con L triangular inferior.

Llamada.- CHOLFOR(MATL, n , rhsol).

Parámetros de entrada.- [1] $n \in \mathbb{Z}$: dimensión del sistema (n° de ecuaciones). [2] $MATL \in E^{n \times n}$: el elemento (i,j) -ésimo de L se almacena en $MATL[i,j]$ ($i = 1,2,\dots,n; j = 1,2,\dots,i$).

Parámetros de entrada-salida.- $rhsol \in E^n$: a la entrada, el elemento i -ésimo de b (el lado derecho del sistema) se almacena en $rhsol[i]$ ($i = 1,2,\dots,n$); a la salida, $rhsol[i]$ ($i = 2,3,\dots,n$) contiene el elemento i -ésimo de x (la solución del sistema).

Otras consideraciones.- Este algoritmo puede utilizarse sólo, o seguido de CHOLBACK para resolver el sistema $Mx = b$; en este caso, L (MATL) es el resultado del algoritmo CHOLDCMP con M (MAT) como parámetro de entrada (vid. algoritmo FVE).

Descripción.- El funcionamiento y la descripción detallada de este algoritmo pueden encontrarse en Dennis y Schnabel (1983), pp. 47-51.

Algoritmo:

```
(* "Forward substitution" *)
1:  $rhsol[1] \leftarrow rhsol[1] / MATL[1,1]$ .
2: FOR  $i = 2$  TO  $n$  DO
    2.1:  $tmp \leftarrow rhsol[i] - \sum_{j=1}^{i-1} MATL[i,j] \times rhsol[j]$ .
    2.2:  $rhsol[i] \leftarrow tmp / MATL[i,i]$ .
3: RETURN. (* Fin normal algoritmo CHOLFOR *)
```

Algoritmo CHOLBACK

Propósito.- Solución del sistema de n ecuaciones lineales $L^T x = b$, con L triangular inferior.

Llamada.- CHOLBACK(MATL, n , rhsol).

Parámetros de entrada.- [1] $n \in \mathbb{Z}$: dimensión del sistema (n° de ecuaciones). [2] $MATL \in E^{n \times n}$: el elemento (i,j) -ésimo de L se almacena en $MATL[i,j]$ ($i = 1,2,\dots,n; j = 1,2,\dots,i$).

Parámetros de entrada-salida.- $rhsol \in E^n$: a la entrada, el elemento i -ésimo de b (el lado derecho del sistema) se almacena en $rhsol[i]$ ($i = 1,2,\dots,n$); a la salida, $rhsol[i]$ ($i = 2,3,\dots,n$) contiene el elemento i -ésimo de x (la solución del sistema).

Otras consideraciones.- Este algoritmo puede utilizarse sólo, o precedido de CHOLFOR para resolver el sistema $Mx = b$; en este caso, L (MATL) es el resultado del algoritmo CHOLDCMP con M (MAT) como parámetro de entrada (vid. algoritmo FVE).

Descripción.- El funcionamiento y la descripción detallada de este algoritmo pueden encontrarse en Dennis y Schnabel (1983), pp. 47-51.

Algoritmo:

(* "Backsubstitution" *)

1: $rhsol[n] \leftarrow rhsol[n] / MATL[n,n]$.

2: FOR $i = n-1$ DOWNTO 1 DO

 2.1: $tmp \leftarrow rhsol[i] - \sum_{j=i+1}^n MATL[j,i] \times rhsol[j]$.

 2.2: $rhsol[i] \leftarrow tmp / MATL[i,i]$.

3: RETURN. (* Fin normal algoritmo CHOLBACK *)

Algoritmo CGAMMA

Propósito.- Cálculo de las matrices de autocovarianzas $\Gamma(k)$ ($k = 0,1,\dots,p$) y de covarianzas cruzadas $\Gamma_{wa}(k)$ ($k = 0,-1,\dots,-q$), según [2.2.7]-[2.2.14] y [2.2.4]-[2.2.6], respectivamente.

Llamada.- CGAMMA(PHI, THETA, QQ, m , p , q , GAMWA, gamma, covok).

Parámetros de entrada.- [1] $m \in \mathbb{Z}$: n° series. [2] $p \in \mathbb{Z}$: orden AR. [3] $q \in \mathbb{Z}$: orden MA. [4] $QQ \in E^{m \times m}$: matriz Q (covarianzas del término de ruido cuando $\sigma^2 = 1$); el elemento (i,j) -ésimo de la matriz Q de [2.1.11] se almacena en $QQ[i,j]$ ($i,j = 1,2,\dots,m$). [5] $PHI(k) \in E^{m \times m}$ ($k = 1,2,\dots,p$): matrices AR; el elemento (i,j) -ésimo de Φ_k se almacena en $PHI(k)[i,j]$ ($k = 1,2,\dots,p; i,j = 1,2,\dots,m$). [6] $THETA(k) \in E^{m \times m}$ ($k = 1,2,\dots,q$): matrices MA; el elemento (i,j) -ésimo de

Θ_k se almacena en $THETA(k)[i,j]$ ($k = 1,2,\dots,q; i,j = 1,2,\dots,m$).

Parámetros de salida.- [1] $gamma \in E^{m^2(p+1)}$: matrices de autocovarianzas teóricas; el elemento (i,j) -ésimo de $\Gamma(k)$ se almacena en $gamma[m^2k+m(i-1)+j]$ ($k = 0,1,\dots,p; i,j = 1,2,\dots,m$). [2] $GAMWA(k) \in E^{m \times m}$ ($k = 0,1,\dots,q$): matrices de covarianzas cruzadas teóricas; el elemento (i,j) -ésimo de $\Gamma_{wa}(-k)$ se almacena en $GAMWA(k)[i,j]$ ($k = 0,1,\dots,q; i,j = 1,2,\dots,m$). [3] $covok \in \{TRUE, FALSE\}$: cuando el sistema lineal [2.2.12] es resuelto sin problemas, se devuelve $covok = TRUE$; en caso contrario, se devuelve $covok = FALSE$ (esta situación indicaría que los parámetros AR están en o cerca de la no estacionariedad).

Almacenamiento intermedio.- [1] $MATPHI \in E^{(m^2(p+1)) \times (m^2(p+1))}$: matriz del sistema lineal [2.2.12]; el elemento (i,j) -ésimo de $\Phi = F_1 + F_2$ (vid. [2.2.13]-[2.2.14]) se almacena en $MATPHI[i,j]$ [$i,j = 1,2,\dots,m^2(p+1)$]. [2] $indx \in Z^{m^2(p+1)}$: vector de números enteros requerido por las rutinas de solución de sistemas lineales (vid. algoritmos **LUDCMP** y **LUSOL**).

Otras consideraciones.- [1] El vector $(m^2(p+1))$ ω del sistema [2.1.12] se calcula (vid. [2.2.10]) y almacena en $gamma$ y pasa como parámetro de entrada-salida al algoritmo **LUSOL**, que devuelve en $gamma$ los elementos de las matrices de autocovarianzas teóricas; por tanto, no se requiere almacenamiento adicional para ω (vid. algoritmo **LUSOL**). [2] Para instrumentar en *FORTRAN*, *Pascal* o *C* el almacenamiento de las matrices $\Gamma_{wa}(-k)$, Φ_k y Θ_k , vid. la introducción a este apéndice.

Descripción (vid. apartado 2.2.1).- En primer lugar, el algoritmo calcula recursivamente las matrices de covarianzas cruzadas $\Gamma_{wa}(-k)$ según [2.2.4] y [2.2.6]. A continuación, calcula y almacena los elementos del vector ω y de la matriz Φ para, finalmente, resolver el sistema lineal [2.2.12] mediante dos algoritmos auxiliares (**LUDCMP** y **LUSOL**).

Algoritmo:

(* Asignación de valores iniciales *)

- 1: $covok \leftarrow TRUE$.
- 2: FOR $i = 1$ TO $m^2 \times (p+1)$ DO
 - 2.1: FOR $j = 1$ TO $m^2 \times (p+1)$ DO
 - MATPHI[i,j] \leftarrow 0.0.
 - 2.2: $gamma[i] \leftarrow$ 0.0.
 - 2.3: $indx[i] \leftarrow$ 0.
- 3: FOR $k = 0$ TO q DO
 - 3.1: FOR $i = 1$ TO m DO
 - 3.1.1: FOR $j = 1$ TO m DO
 - GAMWA(k)[i,j] \leftarrow 0.0.

(* Cálculo de las $q+1$ matrices $(m \times m)$ $\Gamma_{wa}(k)$ ($k = 0, -1, \dots, -q$); vid. [2.2.4]-[2.2.6] *)

4: $GAMWA(0) \leftarrow QQ$.

5: FOR $k = 1$ TO q DO

5.1: FOR $i = 1$ TO m DO

5.1.1: FOR $j = 1$ TO m DO

5.1.1.1: $sum \leftarrow - \sum_{h=1}^m THETA(k)[i,h] \times QQ[h,j]$.

5.1.1.2: FOR $l = 1$ TO p DO

5.1.1.2.1: IF $(k-l \geq 0)$ THEN

$sum \leftarrow sum + \sum_{h=1}^m PHI(l)[i,h] \times GAMWA(k-l)[h,j]$.

5.1.1.3: $GAMWA(k)[i,j] \leftarrow sum$.

(* Cálculo y almacenamiento del vector ω en $gamma$ y de la matriz Φ en $MATPHI$; vid. [2.2.12]-[2.2.14] *)

6: FOR $k = 0$ TO p DO

6.1: FOR $i = 1$ TO m DO

6.1.1: FOR $j = 1$ TO m DO

6.1.1.1: $l \leftarrow k \times m^2 + m \times (i-1) + j$.

6.1.1.2: IF $(k = 0)$ THEN

$gamma[l] \leftarrow QQ[i,j]$.

6.1.1.3: IF $(k > 0)$ AND $(k \leq q)$ THEN

$gamma[l] \leftarrow gamma[l] - \sum_{h=1}^m THETA(k)[j,h] \times QQ[i,h]$.

6.1.1.4: FOR $ii = k+1$ TO q DO

$gamma[l] \leftarrow gamma[l] - \sum_{h=1}^m THETA(ii)[j,h] \times GAMWA(ii-k)[i,h]$.

6.1.1.5: $MATPHI[l,l] \leftarrow 1.0$.

6.1.1.6: FOR $ii = 1$ TO p DO

6.1.1.6.1: FOR $kk = 1$ TO m DO

6.1.1.6.1.1T: IF $(k \geq ii)$ THEN

$ll \leftarrow (k-ii) \times m^2 + (i-1) \times m + kk$

6.1.1.6.1.1E: ELSE

$ll \leftarrow (ii-k) \times m^2 + (kk-1) \times m + i$.

6.1.1.6.1.2: $MATPHI[l,ll] \leftarrow MATPHI[l,ll] - PHI(ii)[j,kk]$.

(* Solución del sistema [2.2.12] (nótese que sólo se necesitan las autocovarianzas cuando hay parte AR) *)

7: IF $(p > 0)$ THEN

(* Descomposición LU de la matriz $MATPHI$; vid. algoritmo LUDCMP *)

7.1: CALL LUDCMP($MATPHI$, $m^2 \times (p+1)$, $indx$, $covok$).

7.2: IF (NOT $covok$) THEN

(* Matriz $MATPHI$ singular \Rightarrow fin anormal algoritmo CGAMMA *)

RETURN.

(* Solución del sistema; vid. algoritmo LUSOL *)

7.3: CALL LUSOL($MATPHI$, $m^2 \times (p+1)$, $indx$, $gamma$).

8: RETURN. (* Fin normal algoritmo CGAMMA *)

Algoritmo CXI

Propósito.- [1] Cálculo y premultiplicación por \mathbf{R} (vid. [2.2.22]) de las matrices $\mathbf{\Xi}_k$ según [2.3.8] ($k = 0, 1, \dots, r^*$). [2] Cálculo del límite superior r^* de la secuencia ($\mathbf{\Xi}_k = \mathbf{0}$ para $k > r^*$). [3] Detección de no invertibilidad.

Llamada.- CXI(THETA, Q1INV, RXI, m, n, q, r*, δ , xiok).

Parámetros de entrada.- [1] $m \in \mathbb{Z}$: n° series. [2] $n \in \mathbb{Z}$: n° observaciones. [3] $q \in \mathbb{Z}$: orden MA. [4] THETA(k) $\in E^{m \times m}$ ($k = 1, 2, \dots, q$): matrices MA; el elemento (i, j)-ésimo de $\mathbf{\Theta}_k$ se almacena en THETA(k)[i, j] ($k = 1, 2, \dots, q$; $i, j = 1, 2, \dots, m$). [5] Q1INV $\in E^{m \times m}$: matriz \mathbf{R} de [2.2.22] (factor de Cholesky inverso de \mathbf{Q}); el elemento (i, j)-ésimo de \mathbf{R} se almacena en Q1INV[i, j] ($i = 1, 2, \dots, m$; $j = 1, 2, \dots, i$; nótese que \mathbf{R} es triangular inferior). [6] $\delta \in E$: tolerancia para determinar la convergencia de la sucesión de matrices $\mathbf{\Xi}_i$; vid. [2.3.44].

Parámetros de salida.- [1] RXI(k) $\in E^{m \times m}$ ($k = 0, 1, \dots, r^*$): el elemento (i, j)-ésimo de $\mathbf{R}\mathbf{\Xi}_k$ se almacena en RXI(k)[i, j] ($k = 0, 1, \dots, r^*$; $i, j = 1, 2, \dots, m$); vid. **Descripción**. [2] $r^* \in \mathbb{Z}$: límite superior de la secuencia de matrices $\mathbf{\Xi}_k$ ($\mathbf{\Xi}_k = \mathbf{0}$ para $k > r^*$). [3] xiok $\in \{\text{TRUE}, \text{FALSE}\}$: indicador de invertibilidad (xiok = TRUE) / no invertibilidad (xiok = FALSE).

Almacenamiento intermedio.- Por motivos de claridad, para realizar las premultiplicaciones de cada matriz $\mathbf{\Xi}_k$ calculada por la matriz \mathbf{R} , se hace uso de una matriz auxiliar $\mathbf{MTMP} \in E^{m \times m}$.

Otras consideraciones.- [1] Para instrumentar en FORTRAN, Pascal o C el almacenamiento de las matrices $\mathbf{\Theta}_k$ y $\mathbf{\Xi}_k$, vid. la introducción a este apéndice. [2] Dado que el límite superior r^* no se conoce *a priori*, conviene reservar almacenamiento suficiente para, en el peor de los casos (procesos casi no invertibles), poder calcular las matrices $\mathbf{\Xi}_k$ hasta $k = n - 1$. [3] El algoritmo está diseñado para que se interrumpa si se satisface la condición [2.3.46]. No obstante, dicha interrupción puede ignorarse eliminando las instrucciones 7.4, 7.5, 7.6.3.5 y 7.6.3.6; esto permitiría evaluar la FVE fuera de la región de invertibilidad, aunque con el riesgo de problemas numéricos (overflow).

Descripción (vid. apartado 2.3.2).- El algoritmo calcula recursivamente las matrices $\mathbf{\Xi}_k$ según [2.3.8] y, simultáneamente, va actualizando un índice r^* que indica el punto a partir del cual pueden considerarse despreciables dichas matrices. El cálculo de r^* se realiza en base al criterio de convergencia [2.3.44], teniendo en cuenta la posible presencia de matrices $\mathbf{\Theta}_i = \mathbf{0}$ para $i < q$. Si se desea evaluar la FVE, debe asignarse a δ un valor negativo; un valor positivo de δ implica una evaluación aproximada de la función de verosimilitud, tanto más precisa cuanto más pequeño sea δ . El algoritmo también comprueba la condición [2.3.46] y devuelve xiok = FALSE si dicha condición se cumple, indicando no invertibilidad del proceso considerado. Finalmente, por motivos

de conveniencia computacional, cada matriz $\tilde{\Xi}_k$ ($k = 0, 1, \dots, r^*$) se devuelve premultiplicada por R (vid. [2.3.9], [2.3.10] y [2.3.19]).

Algoritmo:

(* Asignación de valores iniciales *)

1: $xiok \leftarrow \text{TRUE}$.

2: $delta \leftarrow \text{FALSE}$.

3: $max \leftarrow 0.0$.

4: FOR $k = 0$ TO $n-1$ DO

4.1: FOR $i = 1$ TO m DO

4.1.1: FOR $j = 1$ TO m DO

$RXI(k)[i,j] \leftarrow 0.0$.

5: FOR $i = 1$ TO m DO

$RXI(0)[i,i] \leftarrow 1.0$.

(* Actualización del índice r^* y evaluación de las matrices $\tilde{\Xi}_{r^*}$ en $RXI(r^*)$ *)

6: $r^* \leftarrow 0$.

7: REPEAT

7.1: $r^* \leftarrow r^* + 1$.

(* Cálculo de $\tilde{\Xi}_{r^*}$ en $RXI(r^*)$; vid. [2.3.8] *)

7.2: FOR $j = 1$ TO q DO

7.2.1: IF ($r^* - j \geq 0$) THEN

7.2.1.1: FOR $ii = 1$ TO m DO

7.2.1.1.1: FOR $jj = 1$ TO m DO

$RXI(r^*) \leftarrow RXI(r^*) + \sum_{h=1}^m THETA(j)[ii,h] \times RXI(r^*-j)[h,jj]$.

(* Lado izquierdo de [2.3.44] y [2.3.46] *)

7.3: $s2 \leftarrow \sum_{i=1}^m \sum_{j=1}^m |RXI(r^*)[i,j]|$.

(* Actualización del lado derecho de [2.3.46] *)

7.4: IF ($r^* \leq q$) THEN

$max \leftarrow max + s2$.

(* Comprobación de [2.3.46] *)

7.5: IF ($s2 > max$) THEN

7.5.1: $xiok \leftarrow \text{FALSE}$.

7.5.2: RETURN. (* No invertibilidad \Rightarrow fin anormal algoritmo CXI *)

(* Convergencia aparente: comprobar que no es debida a matrices $\Theta_i = \mathbf{0}$ para $i < q$ *)

7.6: IF ($s2 < \delta$) THEN

7.6.1: $nq \leftarrow 1$.

7.6.2: $delta \leftarrow \text{TRUE}$.

7.6.3: WHILE ($nq \leq q$) AND ($r^* < n-1$) AND ($delta$) DO

7.6.3.1: $nq \leftarrow nq + 1$.
7.6.3.2: $r^* \leftarrow r^* + 1$.
7.6.3.3: FOR $j = 1$ TO q DO
 7.6.3.3.1: IF $(r^* - j \geq 0)$ THEN
 7.6.3.3.1.1: FOR $ii = 1$ TO m DO
 7.6.3.3.1.1.1: FOR $jj = 1$ TO m DO
 $RXI(r^*) \leftarrow RXI(r^*) + \sum_{h=1}^m THETA(j)[ii,h] \times RXI(r^* - j)[h,jj]$.
 (* Lado izquierdo de [2.3.46] *)
7.6.3.4: $s2 \leftarrow \sum_{i=1}^m \sum_{j=1}^m |RXI(r^*)[i,j]|$.
 (* Actualización del lado derecho de [2.3.46] *)
7.6.3.5: IF $(r^* \leq q)$ THEN
 $max \leftarrow max + s2$.
 (* Comprobación de [2.3.46] *)
7.6.3.6: IF $(s2 > max)$ THEN
 7.6.3.6.1: $xiok \leftarrow FALSE$.
 7.6.3.6.2: RETURN. (* No invertibilidad \Rightarrow fin anormal algoritmo CXI *)
7.6.3.7: IF $(s2 > \delta)$ THEN
 $delta \leftarrow FALSE$.
 (* END WHILE 7.6.3 *)
7.6.4: IF $delta$ THEN
 $r^* \leftarrow r^* - nq$. (* Convergencia efectiva *)
 (* END IF 7.6 *)
7U: UNTIL $(delta)$ OR $(r^* = n - 1)$.
 (* Premultiplicar cada \mathbf{Z}_k por \mathbf{R} y sobrescribir $RXI(k)$ *)
8: FOR $k = 0$ TO r^* DO
 8.1: FOR $i = 1$ TO m DO
 8.1.1: FOR $j = 1$ TO m DO
 $MTMP[i,j] \leftarrow \sum_{h=1}^i Q1INV[i,h] \times RXI(k)[h,j]$.
 8.2: $RXI(k) \leftarrow MTMP$.
9: RETURN. (* Fin normal algoritmo CXI *)

Algoritmo CRES

Propósito.- Evaluación *analíticamente exacta*, según [2.3.42], del vector de residuos correspondientes a la muestra y los valores numéricos de los parámetros considerados, a partir de cantidades previamente evaluadas por el algoritmo FVE.

Llamada.- CRES($RXI, Q1, MATM, MATL, lambda, m, n, p, q, g, r^*, res$).

Parámetros de entrada.- [1] $m \in \mathbb{Z}$: n° series. [2] $n \in \mathbb{Z}$: n° observaciones. [3] $p \in \mathbb{Z}$: orden AR. [4] $q \in \mathbb{Z}$: orden MA. [5] $g \in \mathbb{Z}$: máximo(p , q). [6] $r^* \in \mathbb{Z}$: índice tal que $\tilde{\mathbf{z}}_k = \mathbf{0}$ para $k > r^*$ (vid. algoritmo CXI). [7] $RXI(k) \in E^{m \times m}$ ($k = 0, 1, \dots, r^*$): matrices $R\tilde{\mathbf{z}}_k$ (vid. algoritmo CXI); el elemento (i, j)-ésimo de $R\tilde{\mathbf{z}}_k$ se almacena en $RXI(k)[i, j]$ ($k = 0, 1, \dots, r^*$; $i, j = 1, 2, \dots, m$). [8] $Q1 \in E^{m \times m}$: factor de Cholesky de $\mathbf{Q} = \mathbf{Q}_1 \mathbf{Q}_1^T$; el elemento (i, j)-ésimo de \mathbf{Q}_1 se almacena en $Q1[i, j]$ ($i = 1, 2, \dots, m$; $j = 1, 2, \dots, i$; nótese que \mathbf{Q}_1 es triangular inferior). [9] $MATM \in E^{gm \times gm}$: factor de Cholesky de $(\mathbf{V}_1 \boldsymbol{\Omega} \mathbf{V}_1^T) = \mathbf{M} \mathbf{M}^T$ (vid. [FVE.6]); el elemento (i, j)-ésimo de \mathbf{M} se almacena en $MATM[i, j]$ ($i = 1, 2, \dots, m$; $j = 1, 2, \dots, i$; nótese que \mathbf{M} es triangular inferior). [10] $MATL \in E^{gm \times gm}$: factor de Cholesky de $[\mathbf{I}_{gm} + \mathbf{M}^T \mathbf{H}^T \mathbf{H} \mathbf{M}] = \mathbf{L} \mathbf{L}^T$ (vid. [FVE.4] y [FVE.7]); el elemento (i, j)-ésimo de \mathbf{L} se almacena en $MATL[i, j]$ ($i = 1, 2, \dots, gm$; $j = 1, 2, \dots, i$; nótese que \mathbf{L} es triangular inferior). [11] $lambda \in E^{gm}$: solución del sistema triangular $\mathbf{L} \boldsymbol{\lambda} = \mathbf{M}^T \tilde{\mathbf{h}}$ (vid. [FVE.8]); el elemento i -ésimo de $\boldsymbol{\lambda}$ se almacena en $lambda[i]$ ($i = 1, 2, \dots, gm$).

Parámetros de entrada-salida.- $res \in E^{mn}$: a la entrada, $res[i + (t-1)m]$ contiene el elemento i -ésimo del vector $\boldsymbol{\eta}_t$ ($t = 1, 2, \dots, n$; $i = 1, 2, \dots, m$; vid. [2.3.22], [2.3.23] y [2.3.2]); a la salida, $res[i + (t-1)m]$ contiene el elemento i -ésimo del vector de residuos $\hat{\boldsymbol{a}}_t$ ($t = 1, 2, \dots, n$; $i = 1, 2, \dots, m$; vid. [2.3.42] y [2.1.16]).

Almacenamiento intermedio.- Un vector auxiliar $vtmp \in E^m$.

Otras consideraciones.- Para instrumentar en *FORTRAN*, *Pascal* o *C* el almacenamiento de las matrices $R\tilde{\mathbf{z}}_k$, vid. la introducción a este apéndice.

Descripción (vid. apartado 2.3.1).- El algoritmo recibe las matrices $\tilde{\mathbf{z}}_k$ (que forman parte de $\mathbf{D}_{\boldsymbol{\theta}, n}^{-1}$; vid. [2.1.23]) y el vector $\hat{\boldsymbol{a}}_0$ premultiplicados por \mathbf{R} . Por ello, se evalúa en primer lugar la expresión [2.3.42] premultiplicada por \mathbf{R} y, a continuación, el resultado se premultiplica por la inversa de dicha matriz (que entra en el algoritmo en $Q1$) para obtener el vector de residuos evaluado según [2.3.42]. El algoritmo hace uso de la rutina **CHOLBACK** para resolver en \mathbf{c} el sistema triangular $\mathbf{L}^T \mathbf{c} = \boldsymbol{\lambda}$.

Algoritmo:

(* Cálculo de \mathbf{c} (en $lambda$) tal que $\mathbf{L}^T \mathbf{c} = \boldsymbol{\lambda}$ ($lambda$ es sobrescrito; vid. algoritmo **CHOLBACK**) *)

1: **CHOLBACK**($MATL$, $g \times m$ $lambda$).

(* Cálculo de $\mathbf{d} = \mathbf{M} \mathbf{c}$ ($lambda$ se sobrescribe gradualmente para almacenar \mathbf{d}) *)

2: FOR $i = m \times g$ DOWNTO 1 DO

2.1: $s1 \leftarrow \sum_{j=1}^i MATM[i, j] \times lambda[j]$.

2.2: $lambda[i] \leftarrow s1$.

(* Evaluación de [2.3.42] premultiplicada por \mathbf{R} (res se sobrescribe gradualmente) *)

3: FOR $i := 1$ TO n DO
 3.1: FOR $j = 1$ TO i DO
 3.1.1: IF $(i-j \leq r^*)$ AND $(j \leq g)$ THEN
 3.1.1.1: FOR $jj = 1$ TO m DO

$$res[jj+(i-1) \times m] \leftarrow res[jj+(i-1) \times m] - \sum_{h=1}^m RXI(i-j)[jj,h] \times lambda[h+(j-1) \times m].$$

 (* Premultiplicar el resultado anterior por $Q1$ para obtener los residuos en res *)
 4: FOR $j = 1$ TO n DO
 4.1: FOR $i = 1$ TO m DO

$$vtmp[i] \leftarrow \sum_{h=1}^i Q1[i,h] \times res[h+(j-1) \times m].$$

 4.2: FOR $i = 1$ TO m DO

$$res[i+(j-1) \times m] \leftarrow vtmp[i].$$

 5: RETURN. (* Fin normal algoritmo CRES *)

Algoritmo FVE

Propósito.- Evaluación del logaritmo de la *FVE* o de la *FVA* de un proceso $ARMA(p,q)$ multivariante, haciendo uso de las expresiones [2.2.42], [2.3.36] y [2.3.37].

Llamada.- $FVE(PHI, THETA, QQ, mu, w, a, m, p, q, g, n, \delta, atf, f1, f2, \sigma2, logfv, retcd, epsmaq)$.

Parámetros de entrada.- [1] $m \in \mathbb{Z}$: n° de series (dimensión del modelo). [2] $p \in \mathbb{Z}$: orden AR. [3] $q \in \mathbb{Z}$: orden MA. [4] $g \in \mathbb{Z}$: máximo(p, q). [5] $n \in \mathbb{Z}$: n° observaciones. [6] $PHI(k) \in E^{m \times m}$ ($k = 1, 2, \dots, p$): matrices AR; el elemento (i, j) -ésimo de Φ_k se almacena en $PHI(k)[i, j]$ ($k = 1, 2, \dots, p$; $i, j = 1, 2, \dots, m$). [7] $THETA(k) \in E^{m \times m}$ ($k = 1, 2, \dots, q$): matrices MA; el elemento (i, j) -ésimo de Θ_k se almacena en $THETA(k)[i, j]$ ($k = 1, 2, \dots, q$; $i, j = 1, 2, \dots, m$). [8] $QQ \in E^{m \times m}$: matriz Q de [2.1.11] (en general, matriz de covarianzas del término de ruido); el elemento (i, j) -ésimo de Q se almacena en $QQ[i, j]$ ($i = 1, 2, \dots, m$; $j = i, i+1, \dots, m$; tan sólo se utiliza el triángulo superior y la diagonal principal). [9] $mu \in E^m$: vector de medias; el elemento i -ésimo de μ se almacena en $mu[i]$ ($i = 1, 2, \dots, m$). [10] $w \in E^{mn}$: vector de observaciones (series transformadas y diferenciadas); la observación t -ésima de la i -ésima serie (elemento i -ésimo de w_t) se almacena en $w[i+(t-1)m]$ ($t = 1, 2, \dots, n$; $i = 1, 2, \dots, m$). [11] $\sigma2 \in E$: factor σ^2 en [2.1.11] (en general, $\sigma2 = 1.0$). [12] $\delta \in E$: tolerancia para determinar la convergencia de la sucesión de matrices Ξ_t (vid. algoritmo CXI). [13] $atf \in \{TRUE, FALSE\}$: si se desea calcular el vector de residuos, debe entrar $atf = TRUE$; en caso contrario, debe entrar $atf = FALSE$. [14] $epsmaq \in E$: épsilon máquina (vid. algoritmo CEPSMQ en el Apéndice A.3).

Parámetros de salida.- [1] $a \in E^{mn}$: vector de residuos (sólo cuando $atf = TRUE$); $a[i+(t-1)m]$ contiene el elemento i -ésimo del vector \hat{a}_t ($t = 1, 2, \dots, n$; $i = 1, 2, \dots, m$). [2] $f1 \in E$: valor de la

forma cuadrática [2.3.36] (raíz m -ésima de [3.1.6]). [3] $f2 \in E$: valor de la expresión [3.1.7]. [3] $\logfv \in E$: logaritmo de la FVE (cuando $\delta < 0$) o de la FVA (cuando $\delta > 0$). [4] $retcd \in Z$: indicador de la causa de terminación del algoritmo; puede tomar uno de los siguientes valores:

- 0: fin normal (evaluación de la función de verosimilitud sin problemas);
- 1: matriz Q no definida positiva;
- 2: sistema [2.2.12] singular (imposibilidad de calcular las autocovarianzas teóricas);
- 3: matriz $(V_1 \Omega V_1^T)$ no definida positiva (\Rightarrow proceso no estacionario);
- 4: proceso no invertible (vid. algoritmo CXI);
- 5: matriz $[I_{gm} + M^T H^T H M]$ no definida positiva.

Almacenamiento intermedio.- [1] $Q1 \in E^{m \times m}$: factor de Cholesky de Q . [2] $Q1INV \in E^{m \times m}$: factor de Cholesky inverso de Q (matriz R de [2.2.22]). [3] $gamma \in E^{m^2(p+1)}$: matrices de autocovarianzas teóricas; el elemento (i,j) -ésimo de $\Gamma(k)$ se almacena en $gamma[m^2k + m(i-1) + j]$ ($k = 0, 1, \dots, p$; $i, j = 1, 2, \dots, m$). [4] $GAMWA(k) \in E^{m \times m}$ ($k = 0, 1, \dots, q$): matrices de covarianzas cruzadas teóricas; el elemento (i,j) -ésimo de $\Gamma_{wa}(-k)$ se almacena en $GAMWA(k)[i,j]$ ($k = 0, 1, \dots, q$; $i, j = 1, 2, \dots, m$). [5] $RXI(k) \in E^{m \times m}$ ($k = 0, 1, \dots, n-1$): secuencia de matrices $R\Xi_k$; el elemento (i,j) -ésimo de $R\Xi_k$ se almacena en $RXI(k)[i,j]$ ($k = 0, 1, \dots, r^*$; $i, j = 1, 2, \dots, m$). [6] $vtmp1, vtmp2 \in E^m$; $vech \in E^{mg}$; $MTMP1 \in E^{m(p+q) \times mg}$; $MTMP2, MTMP3, OM1 \in E^{mg \times mg}$: vectores y matrices para almacenar resultados intermedios.

Otras consideraciones.- [1] Para instrumentar en FORTRAN, Pascal o C el almacenamiento de las matrices Φ_k , Θ_k , $\Gamma(k)$, $\Gamma_{wa}(-k)$ y $R\Xi_k$, vid. la introducción a este apéndice. [2] Para evaluar la función de verosimilitud, es suficiente almacenar la matriz de covarianzas del término de ruido en QQ con $\sigma^2 = 1.0$; la descomposición que se hace en [2.1.11] sólo es relevante para maximizar dicha función, como se describe en el Capítulo 3 y en el Apéndice A.3.

Descripción (vid. Sección 2.3).- El algoritmo empieza calculando, haciendo uso del algoritmo CHOLDCMP, el determinante $|Q|$ y el factor de Cholesky Q_1 de la matriz Q , así como la matriz $R = Q_1^{-1}$ de [2.2.22]. En segundo lugar, haciendo uso del algoritmo CGAMMA, se calculan las matrices de autocovarianzas $\Gamma(k)$ ($k = 0, 1, \dots, p$) y de covarianzas cruzadas $\Gamma_{wa}(-k)$ ($k = 0, 1, \dots, q$). A continuación, se evalúa la matriz $(V_1 \Omega V_1^T)$, según [2.3.38]-[2.3.40], y se obtiene su descomposición de Cholesky con el algoritmo CHOLDCMP (matriz M de [2.3.33]). Seguidamente, haciendo uso del algoritmo CXI, se evalúa la secuencia de matrices $R\Xi_k$ ($k = 0, 1, \dots, r^*$) y el límite superior r^* de dicha secuencia. El algoritmo prosigue evaluando el vector η según [2.3.22]-[2.3.23] y [2.3.2]. A continuación, se evalúa el vector $M^T \tilde{h}$ haciendo uso de [2.3.19]. Seguidamente, haciendo uso de [2.3.9]-[2.3.10], se calculan los elementos de la matriz $D = [I_{gm} + M^T H^T H M]$, así como su determinante $|D|$ y su factor de Cholesky L (con el algoritmo

CHOLDCMP), para, posteriormente, resolver el sistema triangular $L\lambda = M^T\tilde{h}$ con el algoritmo **CHOLFOR**. El algoritmo devuelve las expresiones $f1 = \eta^T\eta - \lambda^T\lambda$, $f2 = |Q| \times |D|^{1/n}$ y $logfv$ evaluada según [3.1.1]. Por último, si entró $atf = \text{TRUE}$, se evalúa el vector de residuos haciendo uso del algoritmo **CRES**.

Algoritmo:

(* Asignación de valores iniciales *)

1: $f1 \leftarrow 0.0$.

2: $f2 \leftarrow 0.0$.

3: $logfv \leftarrow 0.0$.

4: $retcd \leftarrow 0$.

(* Cálculo de R (en $Q1$) y de $|Q|$ (en $detq$) *)

5: $Q1 \leftarrow QQ$.

6: **CHOLDCMP**($Q1, m, d1, d2, ok, epsmaq$). (* Factor de Cholesky de Q en $Q1$ *)

7: IF (NOT ok) THEN

 (* Matriz Q no definida positiva *)

 7.1: $retcd \leftarrow 1$.

 7.2: RETURN. (* Fin anormal algoritmo **FVE** *)

8: FOR $i = 1$ TO m DO (* Factor de Cholesky inverso de Q en $Q1INV$ *)

 8.1: FOR $j = 1$ TO m DO

$vtmp1[j] \leftarrow 0.0$.

 8.2: $vtmp1[i] \leftarrow 1.0$.

 8.3: **CHOLFOR**($Q1, m, vtmp1$).

 8.4: FOR $j = i$ TO m DO

$Q1INV[j,i] \leftarrow vtmp1[j]$.

9: $detq \leftarrow d1 \times 2.0^{d2}$.

(* Cálculo de las autocovarianzas y de las covarianzas cruzadas *)

10: **CGAMMA**($PHI, THETA, QQ, m, p, q, GAMWA, gamma, ok$).

11: IF (NOT ok) THEN

 (* Matriz del sistema [2.1.12] singular *)

 11.1: $retcd \leftarrow 2$.

 11.2: RETURN. (* Fin anormal algoritmo **FVE** *)

(* Cálculo de M [factor de Cholesky de $(V_1\Omega V_1^T)$] en $OM1$ y verificación de estacionariedad *)

12: FOR $i = 1$ TO $m \times g$ DO

 12.1: FOR $j = 1$ TO $m \times g$ DO

$OM1[i,j] \leftarrow 0.0$.

13: FOR $i = 1$ TO $m \times (p+q)$ DO (* Cálculo de ΩV_1^T en $MTMP1$ según [2.3.39]-[2.3.40] *)

 13.1: FOR $j = 1$ TO $m \times g$ DO

$MTMP1[i,j] \leftarrow 0.0.$

14: FOR $i = 1$ TO p DO (* Evaluación de [2.3.39] *)

14.1: FOR $j = 1$ TO g DO

14.1.1: FOR $k = j-i$ TO $p-i$ DO

14.1.1.1: FOR $ii = 1$ TO m DO

14.1.1.1.1: FOR $jj = 1$ TO m DO

14.1.1.1.1.1: $sum \leftarrow 0.0.$

14.1.1.1.1.2: FOR $kk = 1$ TO m DO

14.1.1.1.1.2.1T: IF $(k \geq 0)$ THEN

$l \leftarrow k \times m^2 + (ii-1) \times m + kk$

14.1.1.1.1.2.1E: ELSE

$l \leftarrow -k \times m^2 + (kk-1) \times m + ii.$

14.1.1.1.1.2.2: $sum \leftarrow sum + gamma[l] \times PHI(p-k-i+j)[jj,kk].$

14.1.1.1.1.3: $fil \leftarrow ii + (i-1) \times m.$

14.1.1.1.1.4: $col \leftarrow jj + (j-1) \times m.$

14.1.1.1.1.5: $MTMP1[fil,col] \leftarrow MTMP1[fil,col] + sum.$

(* END FOR 14.1.1.1.1 *)

14.1.2: FOR $k = j-i$ TO $q-i$ DO

14.1.2.1: IF $(-q+p+k \leq 0)$ THEN

14.1.2.1.1: FOR $ii = 1$ TO m DO

14.1.2.1.1.1: FOR $jj = 1$ TO m DO

14.1.2.1.1.1.1: $sum \leftarrow \sum_{kk=1}^m GAMWA(q-p-k)[ii,kk] \times THETA(q-k-i+j)[jj,kk].$

14.1.2.1.1.1.2: $fil \leftarrow ii + (i-1) \times m.$

14.1.2.1.1.1.3: $col \leftarrow jj + (j-1) \times m.$

14.1.2.1.1.1.4: $MTMP1[fil,col] \leftarrow MTMP1[fil,col] - sum.$

(* END FOR 14.1.2.1.1 *)

(* END FOR 14.1 *)

15: FOR $i = p+1$ TO $p+q$ DO (* Evaluación de [2.3.40] *)

15.1: FOR $j = 1$ TO g DO

15.1.1: FOR $k = p+j-i$ TO $2 \times p-i$ DO

15.1.1.1: IF $(-q+p-k \leq 0)$ THEN

15.1.1.1.1: FOR $ii = 1$ TO m DO

15.1.1.1.1.1: FOR $jj = 1$ TO m DO

15.1.1.1.1.1.1: $sum \leftarrow \sum_{kk=1}^m GAMWA(q-p+k)[kk,ii] \times PHI(2 \times p-k-i+j)[jj,kk].$

15.1.1.1.1.1.2: $fil \leftarrow ii + (i-1) \times m.$

15.1.1.1.1.1.3: $col \leftarrow jj + (j-1) \times m.$

15.1.1.1.1.1.4: $MTMP1[fil,col] \leftarrow MTMP1[fil,col] + sum.$

(* END FOR 15.1.1.1.1.1 *)

15.1.2: IF $(p-i+j \leq 0)$ THEN

```

15.1.2.1: FOR  $ii = 1$  TO  $m$  DO
    15.1.2.1.1: FOR  $jj = 1$  TO  $m$  DO
        15.1.2.1.1.1:  $sum \leftarrow \sum_{kk=1}^m GAMWA(0)[ii,kk] \times THETA(q+p-i+j)[jj,kk]$ .
        15.1.2.1.1.2:  $fil \leftarrow ii + (i-1) \times m$ .
        15.1.2.1.1.3:  $col \leftarrow jj + (j-1) \times m$ .
        15.1.2.1.1.4:  $MTMP1[fil,col] \leftarrow MTMP1[fil,col] - sum$ .
    (* END FOR 15.1.2.1.1 *)
(* END FOR 15.1 *)
16: FOR  $i = 1$  TO  $g$  DO (* Evaluación de [2.3.38] *)
    16.1: FOR  $j = i$  TO  $g$  DO
        16.1.1: FOR  $k = 0$  TO  $p-i$  DO
            16.1.1.1: FOR  $ii = 1$  TO  $m$  DO
                16.1.1.1.1: FOR  $jj = 1$  TO  $m$  DO
                    16.1.1.1.1.1:  $col \leftarrow jj + (j-1) \times m$ .
                    16.1.1.1.1.2:  $sum \leftarrow \sum_{kk=1}^m PHI(p-k)[ii,kk] \times MTMP1[kk + (k+i-1) \times m, col]$ .
                    16.1.1.1.1.3:  $fil \leftarrow ii + (i-1) \times m$ .
                    16.1.1.1.1.4:  $OM1[fil,col] \leftarrow OM1[fil,col] + sum$ .
                (* END FOR 16.1.1.1.1 *)
            16.1.1.2: FOR  $k = 0$  TO  $q-i$  DO
                16.1.2.1: FOR  $ii = 1$  TO  $m$  DO
                    16.1.2.1.1: FOR  $jj = 1$  TO  $m$  DO
                        16.1.2.1.1.1:  $col \leftarrow jj + (j-1) \times m$ .
                        16.1.2.1.1.2:  $sum \leftarrow \sum_{kk=1}^m THETA(q-k)[ii,kk] \times MTMP1[kk + (k+p+i-1) \times m, col]$ .
                        16.1.2.1.1.3:  $fil \leftarrow ii + (i-1) \times m$ .
                        16.1.2.1.1.4:  $OM1[fil,col] \leftarrow OM1[fil,col] - sum$ .
                    (* END FOR 16.1.2.1.1 *)
            17: CHOLDCMP( $OM1, m \times g, d1, d2, ok, epsmaq$ ).
            18: IF (NOT  $ok$ ) THEN
                (* Matriz  $(V_1 \Omega V_1^T)$  no definida positiva  $\Rightarrow$  proceso no estacionario *)
                18.1:  $retcd \leftarrow 3$ .
                18.2: RETURN. (* Fin anormal algoritmo FVE *)
            (* Cálculo de la secuencia  $R\Xi_k$  ( $k = 0, 1, \dots, r^*$ ) en  $RXI(k)$  *)
            19: CXI( $THETA, Q1INV, RXI, m, n, q, r^*, \delta, ok$ ).
            20: IF (NOT  $ok$ ) THEN
                (* Secuencia  $\Xi_k$  explosiva  $\Rightarrow$  proceso no invertible *)
                20.1:  $retcd \leftarrow 4$ .
                20.2: RETURN. (* Fin anormal algoritmo FVE *)
            (* Evaluación del vector  $\eta$  en  $a$  *)
            21: FOR  $i = 1$  TO  $m \times n$  DO

```

$a[i] \leftarrow 0.0.$
22: FOR $i = 1$ TO n DO (* Cálculo recursivo de los residuos condicionados según [2.3.2] *)
 22.1: FOR $j = 1$ TO m DO
 $vtmp1[j] \leftarrow 0.0.$
 22.2: FOR $j = 1$ TO p DO
 22.2.1: IF $(i-j \geq 1)$ THEN
 22.2.1.1: FOR $ii = 1$ TO m DO
 22.2.1.1.1: $sum \leftarrow \sum_{k=1}^m PHI(j)[ii,k] \times (w[k+(i-j-1) \times m] - mu[k]).$
 22.2.1.1.2: $vtmp1[ii] \leftarrow vtmp1[ii] + sum.$
 22.3: FOR $j = 1$ TO m DO
 $vtmp2[j] \leftarrow 0.0.$
 22.4: FOR $j = 1$ TO q DO
 22.4.1: IF $(i-j \geq 1)$ THEN
 22.4.1.1: FOR $ii = 1$ TO m DO
 22.4.1.1.1: $sum \leftarrow \sum_{k=1}^m THETA(j)[ii,k] \times a[k+(i-j-1) \times m].$
 22.4.1.1.2: $vtmp2[ii] \leftarrow vtmp2[ii] + sum.$
 22.5: FOR $ii = 1$ TO m DO
 $a[ii+(i-1) \times m] \leftarrow (w[ii+(i-1) \times m] - mu[ii]) - vtmp1[ii] + vtmp2[ii].$
 (* END FOR **22** *)
23: FOR $i = 1$ TO n DO (* Evaluación de η en a según [2.3.22] *)
 23.1: FOR $j = 1$ TO m DO
 $vtmp1[j] \leftarrow \sum_{k=1}^j Q1INV[j,k] \times a[k+(i-1) \times m].$
 23.2: FOR $j = 1$ TO m DO
 $a[j+(i-1) \times m] \leftarrow vtmp1[j].$
 (* Cálculo del vector $M^T \tilde{h}$ en $vech$ *)
24: FOR $i = 1$ TO $g \times m$ DO
 $vech[i] \leftarrow 0.0.$
25: FOR $j = 1$ TO g DO (* Evaluación del vector \tilde{h} según [2.3.19] *)
 25.1: FOR $i = 0$ TO $n-j$ DO
 25.1.1: IF $(i \leq r^*)$ THEN
 25.1.1.1: FOR $jj = 1$ TO m DO
 25.1.1.1.1: $sum \leftarrow \sum_{k=1}^m RXI(i)[k,jj] \times a[k+(i+j-1) \times m].$
 25.1.1.1.2: $vech[jj+(j-1) \times m] \leftarrow vech[jj+(j-1) \times m] + sum.$
 (* END FOR **25.1.1.1** *)
26: FOR $i = 1$ TO $m \times g$ DO (* Premultiplicación de \tilde{h} por M^T *)
 26.1: $sum \leftarrow \sum_{k=i}^{m \times g} OM1[k,i] \times vech[k].$
 26.2: $vech[i] \leftarrow sum.$
 (* Almacenamiento temporal de M en $MTMP3$ si $atf = TRUE$ (cálculo de residuos solicitado) *)
27: IF atf THEN

27.1: FOR $i = 1$ TO $m \times g$ DO
 27.1.1: FOR $j = 1$ TO i DO
 $MTMP3[i,j] \leftarrow OM1[i,j]$.
 (* Cálculo de $D = [I_{gm} + M^T H^T H M]$ (en $OM1$) y de $|D|$ (en $detom$) *)
 28: FOR $i = 1$ TO $m \times g$ DO (* Cálculo de $H^T H$ en $MTMP2$ según [2.3.9]-[2.3.10] *)
 28.1: FOR $j = 1$ TO $m \times g$ DO
 $MTMP2[i,j] \leftarrow 0.0$.
 29: FOR $i = 1$ TO g DO (* Evaluación de [2.3.9] *)
 29.1: FOR $k = 0$ TO $n-i$ DO
 29.1.1: IF $(k+i-1 \leq r^*)$ THEN
 29.1.1.1: FOR $ii = 1$ TO m DO
 29.1.1.1.1: FOR $jj = 1$ TO m DO
 29.1.1.1.1.1: $sum \leftarrow \sum_{kk=1}^m RXI(k)[kk,ii] \times RXI(k+i-1)[kk,jj]$.
 29.1.1.1.1.2: $MTMP2[ii+(i-1) \times m, jj] \leftarrow MTMP2[ii+(i-1) \times m, jj] + sum$.
 30: FOR $i = 2$ TO g DO (* Evaluación de [2.3.10] *)
 30.1: FOR $j = 2$ TO i DO
 30.1.1: FOR $ii = 1$ TO m DO
 30.1.1.1: FOR $jj = 1$ TO m DO
 30.1.1.1.1: $sum \leftarrow 0.0$.
 30.1.1.1.2: IF $(n-i+1 \leq r^*)$ AND $(n-j+1 \leq r^*)$ THEN
 $sum \leftarrow \sum_{kk=1}^m RXI(n-i+1)[kk,ii] \times RXI(n-j+1)[kk,jj]$.
 30.1.1.1.3: $MTMP2[ii+(i-1) \times m, jj+(j-1) \times m] \leftarrow MTMP2[ii+(i-2) \times m, jj+(j-2) \times m] - sum$.
 31: FOR $i = 1$ TO $m \times g$ DO (* $H^T H$ es simétrica *)
 31.1: FOR $j = i+1$ TO $m \times g$ DO
 $MTMP2[i,j] \leftarrow MTMP2[j,i]$.
 32: FOR $i = 1$ TO $m \times g$ DO (* Cálculo de $M^T H^T H$ en $MTMP1$ *)
 32.1: FOR $j = 1$ TO $m \times g$ DO
 $MTMP1[i,j] \leftarrow \sum_{k=i}^{m \times g} OM1[k,i] \times MTMP2[k,j]$.
 33: $MTMP2 \leftarrow OM1$. (* Almacenamiento de M en $MTMP2$ *)
 34: FOR $i = 1$ TO $m \times g$ DO (* Cálculo de $[I_{gm} + M^T H^T H M]$ en $OM1$ *)
 34.1: FOR $j = i$ TO $m \times g$ DO
 34.1.1: $OM1[i,j] \leftarrow \sum_{k=i}^{m \times g} MTMP1[i,k] \times MTMP2[k,j]$.
 34.1.2: $OM1[j,i] \leftarrow OM1[i,j]$.
 34.2: $OM1[i,i] \leftarrow 1.0 + OM1[i,i]$.
 35: **CHOLDCMP**($OM1, m \times g, d1, d2, ok, epsmaq$).
 36: IF (NOT ok) THEN
 (* Matriz $[I_{gm} + M^T H^T H M]$ no definida positiva *)
 36.1: $retcd \leftarrow 5$.
 36.2: RETURN. (* Fin anormal algoritmo **FVE** *)

37: $detom \leftarrow d1 \times 2.0^{d2}$.

(* Cálculo de la forma cuadrática [2.3.36] en $f1$ *)

38: $s1 \leftarrow \sum_{i=1}^{m \times n} (a[i])^2$. (* $\eta^T \eta$ *)

39: **CHOLFOR**($OM1$, $m \times g$, $vech$). (* Solución del sistema $L\lambda = M^T \tilde{h}$ *)

40: $s2 \leftarrow \sum_{i=1}^{m \times g} (vech[i])^2$. (* $\lambda^T \lambda$ *)

41: $f1 \leftarrow s1 - s2$.

(* Cálculo de la expresión [3.1.7] en $f2$ *)

42: $f2 \leftarrow \log(detom) / n$.

43: $f2 \leftarrow detq \times \exp(f2)$.

(* Cálculo del logaritmo de la función de verosimilitud en $logfv$ (vid. [3.1.1] y [3.1.3]) *)

44: $logfv \leftarrow -0.5 \times (n \times m \times \log(2.0 \times \pi \times \sigma^2)) + n \times \log(detq) + \log(detom) + f1 / \sigma^2$.

(* Evaluación del vector de residuos \hat{a} si $atf = \text{TRUE}$ *)

45: IF atf THEN

CRES(RXI , $Q1$, $MTMP3$, $OM1$, $vech$, m , n , p , q , g , r^* , a).

46: RETURN. (* Fin normal algoritmo **FVE** *)

APENDICE A.3

Codificación del algoritmo [MAXFV]

Introducción

En este apéndice se describe en forma de *pseudocódigo* un algoritmo para maximizar la *FVE* (o la *FVA*) de procesos ARMA multivariantes, haciendo uso de las ideas expuestas en el Capítulo 3 y del algoritmo [FVE], descrito en el Capítulo 2 y codificado en el Apéndice A.2. En dicho apéndice puede encontrarse también una descripción detallada de las normas seguidas para la redacción del *pseudocódigo*.

La Figura A.3.1 representa el diagrama de dependencias y llamadas entre los módulos que componen el algoritmo de estimación (algoritmo **MAXFV**).

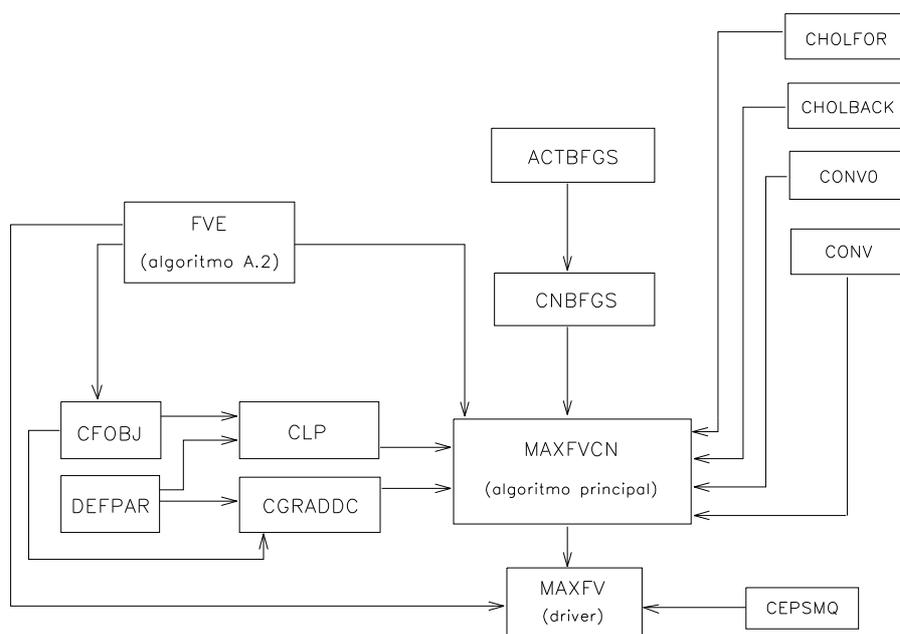


Fig. A.3.1: Diagrama de dependencias del algoritmo **MAXFV**.

El algoritmo **MAXFV** actúa tan sólo como intermediario (*driver*) entre un supuesto programa principal y el algoritmo de estimación **MAXFVCN**. También realiza algunos cálculos auxiliares (como el del ϵ máquina, mediante una llamada al algoritmo **CEPSMQ**). Por tanto, para utilizar el

mecanismo de estimación propuesto en este trabajo, como parte integral de un programa de análisis de series temporales, tan sólo es necesaria una llamada al algoritmo **MAXFV**. El algoritmo **MAXFVCN** (algoritmo principal) contiene el procedimiento de estimación propiamente dicho, descrito en el Capítulo 3. Este procedimiento minimiza la función objetivo [3.1.18] haciendo uso de un método cuasi-Newton basado en la versión factorizada de la fórmula BFGS (algoritmos **CNBFGS** y **ACTBFGS**). El vector gradiente de la función objetivo se evalúa mediante diferencias finitas con el algoritmo **CGRADD**, mientras que la longitud de paso en cada iteración se calcula con el algoritmo **CLP**. Estos dos algoritmos hacen uso de los procedimientos **CFOBJ** (para evaluar la función objetivo [3.1.18]) y **DEFPAR** (para definir el vector de parámetros a estimar). El algoritmo **FVE** del Apéndice A.2 se utiliza directamente en los algoritmos **CFOBJ**, **MAXFVCN** y **MAXFV**. Por último, las rutinas **CONV0** y **CONV** evalúan la convergencia del proceso iterativo, mientras que los algoritmos **CHOLFOR** y **CHOLBACK** (*vid.* Apéndice A.2) se emplean para resolver el sistema de ecuaciones lineales que proporciona la dirección de búsqueda en cada iteración y para invertir la matriz de segundas derivadas en la última iteración.

Algoritmo CEPSMQ

Propósito.- Cálculo del épsilon máquina.

Llamada.- **CEPSMQ**(*epsmaq*).

Parámetros de salida.- *epsmaq* ∈ *E*: épsilon máquina.

Algoritmo:

- 1: *epsmaq* ← 1.0.
- 2: REPEAT
 - epsmaq* ← *epsmaq* / 2.0
- 2U: UNTIL (*epsmaq* + 1.0) = 1.0.
- 3: *epsmaq* ← 2.0 × *epsmaq*.
- 4: RETURN. (* Fin normal algoritmo **CEPSMQ** *)

Algoritmo DEFPAR

Propósito.- Asignar a los parámetros estándar del modelo considerado (Φ_i , Θ_i , μ y Q) los valores numéricos derivados de (i) los valores numéricos de los parámetros que efectivamente se desea

estimar y (ii) las posibles restricciones sobre los parámetros estándar. El algoritmo está codificado contemplando la posibilidad de fijar el valor de cualquier parámetro de las matrices Φ_i , Θ_i , μ y Q , de manera que dicho parámetro no se estime. En el apartado **Otras consideraciones** se describe cómo incorporar otro tipo de restricciones sencillas entre parámetros.

Llamada.- DEFPAR(*npar*, *x*, *m*, *p*, *q*, *PHI*, *IPHI*, *THETA*, *ITHETA*, *mu*, *imu*, *QQ*, *IQQ*).

Parámetros de entrada.- [1] *npar* ∈ Z: n° de parámetros a estimar. [2] *x* ∈ E^{npar} : valores numéricos de los *npar* parámetros que se desea estimar. [3] *m* ∈ Z: n° de series (dimensión del modelo). [4] *p* ∈ Z: orden AR. [5] *q* ∈ Z: orden MA. [6] *IPHI*(*k*) ∈ $Z^{m \times m}$ (*k* = 1,2,...,*p*): si *IPHI*(*k*)[*i,j*] = 1, el elemento (*i,j*)-ésimo de Φ_k es un parámetro a estimar y, por tanto, forma parte del vector de parámetros *x*; en caso contrario, el elemento (*i,j*)-ésimo de Φ_k se mantiene fijo durante la estimación. [7] *ITHETA*(*k*) ∈ $Z^{m \times m}$ (*k* = 1,2,...,*q*): si *ITHETA*(*k*)[*i,j*] = 1, el elemento (*i,j*)-ésimo de Θ_k es un parámetro a estimar y, por tanto, forma parte del vector de parámetros *x*; en caso contrario, el elemento (*i,j*)-ésimo de Θ_k se mantiene fijo durante la estimación. [8] *imu* ∈ Z^m : si *imu*[*i*] = 1, el elemento *i*-ésimo de μ es un parámetro a estimar y, por tanto, forma parte del vector de parámetros *x*; en caso contrario, el elemento *i*-ésimo de μ se mantiene fijo durante la estimación. [9] *IQQ* ∈ $Z^{m \times m}$: si *IQQ*[*i,j*] = 1, el elemento (*i,j*)-ésimo de la matriz *Q* es un parámetro a estimar y, por tanto, forma parte del vector de parámetros *x*; en caso contrario, el elemento (*i,j*)-ésimo de *Q* se mantiene fijo durante la estimación.

Parámetros de entrada-salida.- [1] *PHI*(*k*) ∈ $E^{m \times m}$ (*k* = 1,2,...,*p*): matrices AR; el elemento (*i,j*)-ésimo de Φ_k se almacena en *PHI*(*k*)[*i,j*] (*k* = 1,2,...,*p*; *i,j* = 1,2,...,*m*). [2] *THETA*(*k*) ∈ $E^{m \times m}$ (*k* = 1,2,...,*q*): matrices MA; el elemento (*i,j*)-ésimo de Θ_k se almacena en *THETA*(*k*)[*i,j*] (*k* = 1,2,...,*q*; *i,j* = 1,2,...,*m*). [3] *QQ* ∈ $E^{m \times m}$: matriz *Q* de [2.1.11]; el elemento (*i,j*)-ésimo de *Q* se almacena en *QQ*[*i,j*] (*i* = 1,2,...,*m*; *j* = 1,2,...,*m*). [4] *mu* ∈ E^m : vector de medias; el elemento *i*-ésimo de μ se almacena en *mu*[*i*] (*i* = 1,2,...,*m*).

Otras consideraciones.- Si el único tipo de restricciones consideradas consiste en fijar el valor de algunos parámetros de las matrices Φ_i , Θ_i , μ y *Q*, el algoritmo no requiere modificaciones. En tal caso, los parámetros de entrada-salida [1]-[4] contienen a la entrada, en sus respectivas posiciones, los valores de los parámetros que se mantienen fijos y, a la salida, tanto dichos valores como los de los parámetros que se están estimando (que entran en el vector *x*). Por el contrario, si se contemplan otro tipo de restricciones, éstas deben poder ser resueltas por sustitución y, en tal caso, debe codificarse explícitamente la dependencia de cada componente de las matrices Φ_i , Θ_i , μ y *Q* respecto a los parámetros que se desea estimar, a la vez que todas las referencias a los parámetros *IPHI*, *ITHETA*, *imu* e *IQQ* pueden ser suprimidas de todo el sistema. La presencia de restricciones más complejas implica el uso de un algoritmo de optimización con restricciones (*vid.* Capítulo 3).

Descripción.- El algoritmo coloca, en las posiciones correspondientes a los parámetros de las matrices Φ_i , Θ_i , μ y Q que se están estimando, los valores numéricos de dichos parámetros (contenidos en el vector x) en diversas fases de la estimación. Al mismo tiempo, mantiene tanto las posiciones como los valores numéricos de los parámetros que son fijos (si hubiera alguno). Por supuesto, cualesquiera que sean las restricciones, éstas pueden codificarse explícitamente en cada caso. No obstante, la utilidad de este algoritmo consiste en proporcionar una forma estándar para definir los parámetros de Φ_i , Θ_i , μ y Q que puede emplearse en muchas ocasiones.

Algoritmo:

1: $nn \leftarrow 0$.

(* Definición del vector μ en términos de los parámetros a estimar *)

2: FOR $i = 1$ TO m DO

2.1: IF ($imu[i] = 1$) THEN

2.1.1: $nn \leftarrow nn + 1$.

2.1.2: $mu[i] \leftarrow x[nn]$.

(* Definición de las matrices Φ_k en términos de los parámetros a estimar *)

3: FOR $k = 1$ TO p DO

3.1: FOR $j = 1$ TO m DO

3.1.1: FOR $i = 1$ TO m DO

3.1.1.1: IF ($IPHI(k)[i,j] = 1$) THEN

3.1.1.1.1: $nn \leftarrow nn + 1$.

3.1.1.1.2: $PHI(k)[i,j] \leftarrow x[nn]$.

(* Definición de las matrices Θ_k en términos de los parámetros a estimar *)

4: FOR $k = 1$ TO q DO

4.1: FOR $j = 1$ TO m DO

4.1.1: FOR $i = 1$ TO m DO

4.1.1.1: IF ($ITHETA(k)[i,j] = 1$) THEN

4.1.1.1.1: $nn \leftarrow nn + 1$.

4.1.1.1.2: $THETA(k)[i,j] \leftarrow x[nn]$.

(* Definición de la matriz Q en términos de los parámetros a estimar *)

5: FOR $i = 1$ TO m DO

5.1: FOR $j = i$ TO m DO

5.1.1: IF ($IQQ[i,j] = 1$) OR ($IQQ[j,i] = 1$) THEN

5.1.1.1: $nn \leftarrow nn + 1$.

5.1.1.2: $QQ[i,j] \leftarrow x[nn]$.

5.1.1.3: $QQ[j,i] \leftarrow QQ[i,j]$.

6: RETURN. (* Fin normal algoritmo **DEFPAR** *)

Algoritmo CFOBJ

Propósito.- Evaluar la función objetivo [3.1.18] haciendo uso del algoritmo **FVE** del Apéndice A.2.

Llamada.- **CFOBJ**(*PHI*, *THETA*, *QQ*, *mu*, *w*, *a*, *m*, *p*, *q*, *g*, *n*, $\Pi10$, $\Pi20$, δ , *epsmaq*, $\Pi1$, $\Pi2$, *fobj*).

Parámetros de entrada.- [1] $m \in \mathbb{Z}$: n° de series (dimensión del modelo). [2] $p \in \mathbb{Z}$: orden AR. [3] $q \in \mathbb{Z}$: orden MA. [4] $g \in \mathbb{Z}$: máximo(p, q). [5] $n \in \mathbb{Z}$: n° observaciones. [6] $PHI(k) \in E^{m \times m}$ ($k = 1, 2, \dots, p$): matrices AR. [7] $THETA(k) \in E^{m \times m}$ ($k = 1, 2, \dots, q$): matrices MA. [8] $QQ \in E^{m \times m}$: matriz Q de [2.1.11]. [9] $mu \in E^m$: vector de medias. [10] $w \in E^{mn}$: vector de observaciones. [11] $a \in E^{mn}$: vector auxiliar. [12] $\Pi10 \in E$: valor de la forma cuadrática [2.3.36] (raíz m -ésima de [3.1.6]) en las estimaciones iniciales. [13] $\Pi20 \in E$: valor de la expresión [3.1.7] en las estimaciones iniciales. [14] $\delta \in E$: tolerancia para determinar la convergencia de la sucesión de matrices $\tilde{\Sigma}_i$. [15] *epsmaq* $\in E$: épsilon máquina.

Parámetros de salida.- [1] $\Pi1 \in E$: valor de la forma cuadrática [2.3.36] (raíz m -ésima de [3.1.6]) evaluada en el punto representado por los valores de entrada de Φ_i , Θ_i , μ y Q . [2] $\Pi2 \in E$: valor de la expresión [3.1.7] evaluada en el punto representado por los valores de entrada de Φ_i , Θ_i , μ y Q . [3] *fobj* $\in E$: valor de la función objetivo [3.1.18] evaluada en el punto representado por los valores de entrada de Φ_i , Θ_i , μ y Q ; cuando dicho punto no es admisible, se devuelve *fobj* = 1.0.

Descripción.- El algoritmo evalúa las expresiones [3.1.6] y [3.1.7] haciendo uso del algoritmo **FVE** y devuelve (i) el valor de [3.1.18] cuando el código de retorno del algoritmo **FVE** es 0 (indicando una evaluación correcta) ó (ii) 1.0 en caso contrario (cuando Q no es definida positiva o cuando el proceso es no estacionario o no invertible; *vid.* Apéndice A.2).

Algoritmo:

(* Evaluación de la función de verosimilitud *)

1: **FVE**(*PHI*, *THETA*, *QQ*, *mu*, *w*, *a*, *m*, *p*, *q*, *g*, *n*, δ , FALSE, $\Pi1$, $\Pi2$, 1.0, *tmp*, *retcd*, *epsmaq*).

2T: IF (*retcd* \neq 0) THEN

(* Punto no admisible *)

fobj \leftarrow 1.0

2E: ELSE

(* Evaluación adecuada *)

fobj \leftarrow ($\Pi1 / \Pi10$) ^{m} \times ($\Pi2 / \Pi20$).

3: RETURN. (* Fin normal algoritmo **CFOBJ** *)

Algoritmo CGRADDC

Propósito.- Evaluación mediante diferencias finitas del gradiente de la función objetivo [3.1.18].

Llamada.- CGRADDC(PHI, IPHI, THETA, ITHETA, QQ, IQQ, mu, imu, w, a, m, p, q, g, n, Π10, Π20, δ, epsmaq, x, grad, npar).

Parámetros de entrada.- [1] $m \in \mathbb{Z}$: n° de series (dimensión del modelo). [2] $p \in \mathbb{Z}$: orden AR. [3] $q \in \mathbb{Z}$: orden MA. [4] $g \in \mathbb{Z}$: máximo(p,q). [5] $n \in \mathbb{Z}$: n° observaciones. [6] $PHI(k) \in E^{m \times m}$ ($k = 1, 2, \dots, p$): matrices AR. [7] $IPHI(k) \in Z^{m \times m}$ ($k = 1, 2, \dots, p$) (vid. algoritmo DEFPAR). [8] $THETA(k) \in E^{m \times m}$ ($k = 1, 2, \dots, q$): matrices MA. [9] $ITHETA(k) \in Z^{m \times m}$ ($k = 1, 2, \dots, q$) (vid. algoritmo DEFPAR). [10] $QQ \in E^{m \times m}$: matriz Q de [2.1.11]. [11] $IQQ \in Z^{m \times m}$ (vid. algoritmo DEFPAR). [12] $mu \in E^m$: vector de medias. [13] $imu \in Z^m$ (vid. algoritmo DEFPAR). [14] $w \in E^{mn}$: vector de observaciones. [15] $a \in E^{mn}$: vector auxiliar. [16] $\Pi10 \in E$: valor de la forma cuadrática [2.3.36] (raíz m -ésima de [3.1.6]) en las estimaciones iniciales. [17] $\Pi20 \in E$: valor de la expresión [3.1.7] en las estimaciones iniciales. [18] $\delta \in E$: tolerancia para determinar la convergencia de la sucesión de matrices $\tilde{\Sigma}_i$. [19] $epsmaq \in E$: épsilon máquina. [20] $npar \in \mathbb{Z}$: n° de parámetros a estimar. [21] $x \in E^{npar}$: valores numéricos de los $npar$ parámetros a estimar.

Parámetros de salida.- $grad \in E^{npar}$: vector gradiente de la función objetivo [3.1.18].

Descripción.- El funcionamiento y la descripción detallada de este algoritmo pueden encontrarse en Dennis y Schnabel (1983), pp. 323-5 (vid. también nota [4] al final de la Sección 3.1).

Algoritmo:

- 1: $m13 \leftarrow (epsmaq)^{1/3}$.
- 2: FOR $i = 1$ TO $npar$ DO
 - 2.1: $hi \leftarrow m13 \times \max(|x[i]|, 1.0)$.
 - 2.2: $xtmpi \leftarrow x[i]$.
 - 2.3: $x[i] \leftarrow xtmpi + hi$.
 - 2.4: DEFPAR($npar, x, m, p, q, PHI, IPHI, THETA, ITHETA, mu, imu, QQ, IQQ$).
 - 2.5: CFOBJ($PHI, THETA, QQ, mu, w, a, m, p, q, g, n, \Pi10, \Pi20, \delta, epsmaq, tmp1, tmp2, fp$).
 - 2.6: $x[i] \leftarrow xtmpi - hi$.
 - 2.7: DEFPAR($npar, x, m, p, q, PHI, IPHI, THETA, ITHETA, mu, imu, QQ, IQQ$).
 - 2.8: CFOBJ($PHI, THETA, QQ, mu, w, a, m, p, q, g, n, \Pi10, \Pi20, \delta, epsmaq, tmp1, tmp2, fm$).
 - 2.9: $grad[i] \leftarrow (fp - fm) / (hi + hi)$.
 - 2.10: $x[i] \leftarrow xtmpi$.
 - 2.11: DEFPAR($npar, x, m, p, q, PHI, IPHI, THETA, ITHETA, mu, imu, QQ, IQQ$).
- 3: RETURN. (* Fin normal algoritmo CGRADDC *)

Algoritmo CLP

Propósito.- Calcular en la i -ésima iteración una longitud de paso α_i tal que $F(x_i + \alpha_i d_i) \leq F(x_i) + \lambda \alpha_i \nabla F(x_i)^T d_i$, donde x_i contiene las estimaciones actuales de los parámetros, d_i es la dirección de búsqueda, $\nabla F(x_i)$ es el vector gradiente de la función objetivo [3.1.18] evaluado en x_i (con $\nabla F(x_i)^T d_i < 0$) y $\lambda = 10^{-4}$ (vid. nota [3] al final de la Sección 3.1).

Llamada.- CLP(PHI, IPHI, THETA, ITHETA, QQ, IQQ, mu, imu, w, a, m, p, q, g, n, Pi10, Pi20, delta, epsmaq, npar, xold, fold, gold, dold, xnew, Pi1, Pi2, fnew, maxp, tolp, mxt, retcd).

Parámetros de entrada.- [1] $m \in \mathbb{Z}$: n° de series (dimensión del modelo). [2] $p \in \mathbb{Z}$: orden AR. [3] $q \in \mathbb{Z}$: orden MA. [4] $g \in \mathbb{Z}$: máximo(p, q). [5] $n \in \mathbb{Z}$: n° observaciones. [6] $PHI(k) \in E^{m \times m}$ ($k = 1, 2, \dots, p$): matrices AR. [7] $IPHI(k) \in Z^{m \times m}$ ($k = 1, 2, \dots, p$) (vid. algoritmo DEFPAR). [8] $THETA(k) \in E^{m \times m}$ ($k = 1, 2, \dots, q$): matrices MA. [9] $ITHETA(k) \in Z^{m \times m}$ ($k = 1, 2, \dots, q$) (vid. algoritmo DEFPAR). [10] $QQ \in E^{m \times m}$: matriz Q de [2.1.11]. [11] $IQQ \in Z^{m \times m}$ (vid. algoritmo DEFPAR). [12] $mu \in E^m$: vector de medias. [13] $imu \in Z^m$ (vid. algoritmo DEFPAR). [14] $w \in E^{mn}$: vector de observaciones. [15] $a \in E^{mn}$: vector auxiliar. [16] $Pi10 \in E$: valor de la forma cuadrática [2.3.36] (raíz m -ésima de [3.1.6]) en las estimaciones iniciales. [17] $Pi20 \in E$: valor de la expresión [3.1.7] en las estimaciones iniciales. [18] $\delta \in E$: tolerancia para determinar la convergencia de la sucesión de matrices \mathfrak{Z}_i . [19] $epsmaq \in E$: épsilon máquina. [20] $npar \in \mathbb{Z}$: n° de parámetros a estimar. [21] $xold \in E^{npar}$: valores numéricos de los parámetros en la iteración actual (x_i). [22] $fold \in E$: valor de la función objetivo [3.1.18] en la iteración actual [$F(x_i)$]. [23] $gold \in E^{npar}$: vector gradiente de la función objetivo en la iteración actual [$\nabla F(x_i)$]. [24] $dold \in E^{npar}$: dirección de búsqueda (d_i). [25] $maxp \in E$: máximo tamaño permitido del paso $\alpha_i d_i$. [26] $tolp \in E$: tamaño relativo del paso $\alpha_i d_i$ para el cual se considera que x_{i+1} difiere lo suficientemente poco de x_i como para terminar el algoritmo.

Parámetros de salida.- [1] $xnew \in E^{npar}$: valores actualizados de los parámetros ($x_{i+1} = x_i + \alpha_i d_i$). [2] $Pi1 \in E$: valor de la forma cuadrática [2.3.36] (raíz m -ésima de [3.1.6]) en las nuevas estimaciones. [3] $Pi2 \in E$: valor de la expresión [3.1.7] en las nuevas estimaciones. [4] $fnew \in E$: valor de la función objetivo [3.1.18] en las nuevas estimaciones. [5] $mxt \in \{\text{TRUE}, \text{FALSE}\}$: el algoritmo devuelve $mxt = \text{TRUE}$ cuando el tamaño del paso $\alpha_i d_i$ es el máximo permitido; en caso contrario, devuelve $mxt = \text{FALSE}$. [6] $retcd \in \mathbb{Z}$: indicador de la causa de terminación del algoritmo (código de retorno); puede tomar los valores 0 (fin satisfactorio \Rightarrow nueva iteración) ó 1 (imposible encontrar x_{i+1} suficientemente distinto de $x_i \Rightarrow$ posible convergencia).

Descripción.- El funcionamiento y la descripción detallada de este algoritmo pueden encontrarse en Dennis y Schnabel (1983), pp. 116-29.

Algoritmo:

1: $mxt \leftarrow \text{FALSE}$.

2: $retcd \leftarrow 2$.

3: $\lambda \leftarrow 10^{-4}$.

(* Tamaño de la dirección de búsqueda: $\|d_i\|_2$ *)

4: $sln \leftarrow \|dold\|_2$.

5: IF ($sln > maxp$) THEN

(* Tamaño del paso mayor que el máximo permitido *)

5.1: $dold \leftarrow (maxp / sln) \times dold$.

5.2: $sln \leftarrow maxp$.

6: $slp \leftarrow gold^T dold$. (* $\nabla F(x_i)^T d_i < 0$ *)

7: $rln \leftarrow \max_{1 \leq i \leq npar} (|gold[i]| / \max(|xold[i]|, 1.0))$.

8: $minalpha \leftarrow tolp / rln$. (* $minalpha$ es el menor tamaño de paso permitido *)

9: $alpha \leftarrow 1.0$.

(* Comprobar si la actualización es adecuada; si no, calcular una nueva longitud de paso *)

10: REPEAT

10.1: $xnew \leftarrow xold + alpha \times dold$. (* $x_{i+1} = x_i + \alpha_i d_i$ *)

(* $F(x_{i+1})$ *)

10.2: DEFPAR($npar, xnew, m, p, q, PHI, IPHI, THETA, ITHETA, mu, imu, QQ, IQQ$).

10.3: CFOBJ($PHI, THETA, QQ, mu, w, a, m, p, q, g, n, \Pi10, \Pi20, \delta, epsmaq, \Pi1, \Pi2, fnew$).

10.4A: IF $fnew \leq (fold + \lambda \times alpha \times slp)$ THEN

(* x_{i+1} es una actualización adecuada de las estimaciones *)

10.4A.1: $retcd \leftarrow 0$.

10.4A.2: IF ($alpha = 1.0$) AND ($sln > (0.99 \times maxp)$) THEN

$mxt \leftarrow \text{TRUE}$.

(* Fin satisfactorio algoritmo CLP *)

10.4B: ELSE IF ($alpha < minalpha$) THEN

(* No puede encontrarse x_{i+1} suficientemente distinto de x_i *)

10.4B.1: $retcd \leftarrow 1$.

10.4B.2: $xnew \leftarrow xold$.

10.4B.3: DEFPAR($npar, xnew, m, p, q, PHI, IPHI, THETA, ITHETA, mu, imu, QQ, IQQ$).

10.4B.4: CFOBJ($PHI, THETA, QQ, mu, w, a, m, p, q, g, n, \Pi10, \Pi20, \delta, epsmaq, \Pi1, \Pi2, fnew$).

(* Fin algoritmo CLP *)

10.4C: ELSE (* Reducción de la longitud de paso $alpha$ *)

10.4C.1T: IF ($alpha = 1.0$) THEN

(* Interpolación cuadrática *)

$alphatmp \leftarrow -slp / (2.0 \times (fnew - fold - slp))$.

10.4C.1E: ELSE

(* Interpolación cúbica *)

10.4C.1E.1: $t1 \leftarrow fnew - fold - alpha \times slp$.

10.4C.1E.2: $t2 \leftarrow pfnew - fold - palpha \times slp$.

10.4C.1E.3: $t3 \leftarrow 1.0 / (alpha - palpha)$.

10.4C.1E.4: $a \leftarrow t3 \times (t1 / alpha^2 - t2 / palpha^2)$.

10.4C.1E.5: $b \leftarrow t3 \times (t2 \times alpha / palpha^2 - t1 \times palpha / alpha^2)$.

10.4C.1E.6: $disc \leftarrow b^2 - 3.0 \times a \times slp$.

10.4C.1E.7T: IF ($a = 0.0$) THEN

$alpmatp \leftarrow -slp / (2.0 \times b)$

10.4C.1E.7E: ELSE

$alpmatp \leftarrow (-b + disc^{1/2}) / (3.0 \times a)$.

10.4C.1E.8: IF ($alpmatp > 0.5 \times alpha$) THEN

$alpmatp \leftarrow 0.5 \times alpha$.

(* END ELSE 10.4C.1E *)

10.4C.2: $palpha \leftarrow alpha$.

10.4C.3: $pfnew \leftarrow fnew$.

10.4C.4T: IF ($alpmatp \leq 0.1 \times alpha$) THEN

$alpha \leftarrow 0.1 \times alpha$

10.4C.4E: ELSE

$alpha \leftarrow alpmatp$.

(* END ELSE 10.4.C *)

10U: UNTIL ($retcd < 2$).

11: RETURN (* Fin normal algoritmo CLP *)

Algoritmo ACTBFGS

Propósito.- Dada una matriz $(n \times n)$ L^T triangular superior y dos vectores $(n \times 1)$ u y v , obtener una matriz L_+^T triangular superior tal que $L_+ L_+^T = (L + v u^T)(L^T + u v^T)$. Nótese que $L_+^T = R_+$ (triangular superior) en la descomposición QR de $J_+^T = (L^T + u v^T) = Q_+ R_+$, dado que $J_+ J_+^T = R_+^T Q_+^T Q_+ R_+ = R_+^T R_+$, puesto que la matriz Q_+ es ortogonal.

Llamada.- ACTBFGS(R, u, v, n).

Parámetros de entrada.- [1] $n \in \mathbb{Z}$: dimensión del problema. [2] $u \in E^n$: vector u . [3] $v \in E^n$: vector v .

Parámetros de entrada-salida.- $R \in E^{n \times n}$: el triángulo superior de R contiene la matriz L^T a la entrada y la matriz L_+^T a la salida.

Descripción.- El funcionamiento y la descripción detallada de este algoritmo pueden encontrarse en Dennis y Schnabel (1983), pp. 311-2.

Algoritmo:

1: $k \leftarrow n$.

2: WHILE ($u[k] = 0.0$) AND ($k > 1$) DO

$k \leftarrow k - 1$.

(* Obtener la matriz R_+ en la descomposición $(L^T + uv^T) = Q_+ R_+ *$)

3: FOR $i = k-1$ DOWNTO 1 DO

3.1T: IF ($u[i] = 0.0$) THEN

3.1T.1: FOR $j = i$ TO n DO

3.1T.1.1: $tmp \leftarrow R[i,j]$.

3.1T.1.2: $R[i,j] \leftarrow R[i+1,j]$.

3.1T.1.3: $R[i+1,j] \leftarrow tmp$.

3.1T.2: $u[i] \leftarrow u[i+1]$.

3.1E: ELSE

3.1E.1: $c \leftarrow u[i] / (u[i]^2 + u[i+1]^2)^{1/2}$.

3.1E.2: $s \leftarrow -u[i+1] / (u[i]^2 + u[i+1]^2)^{1/2}$.

3.1E.3: FOR $j = i$ TO n DO

3.1E.3.1: $y \leftarrow R[i,j]$.

3.1E.3.2: $z \leftarrow R[i+1,j]$.

3.1E.3.3: $R[i,j] \leftarrow c \times y - s \times z$.

3.1E.3.4: $R[i+1,j] \leftarrow s \times y + c \times z$.

3.1E.4: $u[i] \leftarrow (u[i]^2 + u[i+1]^2)^{1/2}$.

4: FOR $j = 1$ TO n DO

$R[1,j] \leftarrow R[1,j] + u[1] \times v[j]$.

5: FOR $i = 1$ TO $k-1$ DO

5.1T: IF ($R[i,i] = 0.0$) THEN

5.1T.1: FOR $j = i$ TO n DO

5.1T.1.1: $tmp \leftarrow R[i,j]$.

5.1T.1.2: $R[i,j] \leftarrow R[i+1,j]$.

5.1T.1.3: $R[i+1,j] \leftarrow tmp$.

5.1E: ELSE

5.1E.1: $c \leftarrow R[i,i] / (R[i,i]^2 + R[i+1,i]^2)^{1/2}$.

5.1E.2: $s \leftarrow -R[i+1,i] / (R[i,i]^2 + R[i+1,i]^2)^{1/2}$.

5.1E.3: FOR $j = i$ TO n DO

5.1E.3.1: $y \leftarrow R[i,j]$.

5.1E.3.2: $z \leftarrow R[i+1,j]$.

5.1E.3.3: $R[i,j] \leftarrow c \times y - s \times z$.

5.1E.3.4: $R[i+1,j] \leftarrow s \times y + c \times z$.

6: RETURN. (* Fin normal algoritmo ACTBFGS *)

Algoritmo CNBFGS

Propósito.- Actualizar el factor de Cholesky L (triangular inferior) de la matriz $(n \times n)$ $G = LL^T$ para obtener el factor de Cholesky L_+ de la matriz $G_+ = L_+L_+^T$, donde G_+ se obtiene a partir de G mediante la fórmula BFGS (vid. nota [5] al final de la Sección 3.1).

Llamada.- CNBFGS(*xold*, *gold*, *xnew*, *gnew*, L , n , *epsmaq*, *itm*, *noact*).

Parámetros de entrada.- [1] $n \in \mathbb{Z}$: dimensión del problema. [2] $xold \in E^n$: x_i . [3] $gold \in E^n$: $\nabla F(x_i)$. [4] $xnew \in E^n$: x_{i+1} . [5] $gnew \in E^n$: $\nabla F(x_{i+1})$. [6] *epsmaq* $\in E$: épsilon máquina. [7] $itm \in \mathbb{Z}$: iteración actual (i).

Parámetros de entrada-salida.- [1] $L \in E^{n \times n}$: el triángulo inferior y la diagonal principal de L contienen la matriz L a la entrada y la matriz L_+ a la salida. [2] *noact* $\in \{\text{TRUE}, \text{FALSE}\}$: indicador de si se ha llevado a cabo alguna actualización.

Almacenamiento intermedio.- Cuatro vectores $s, y, u, v \in E^n$.

Descripción.- El funcionamiento y la descripción detallada de este algoritmo pueden encontrarse en Dennis y Schnabel (1983), pp. 198-201 y 356-8.

Algoritmo:

```

1: IF (itm = 1) THEN
    noact  $\leftarrow$  TRUE.
2:  $s \leftarrow xnew - xold$ . (*  $s = x_{i+1} - x_i = \alpha_i d_i$  *)
3:  $y \leftarrow gnew - gold$ . (*  $y = \nabla F(x_{i+1}) - \nabla F(x_i)$  *)
4:  $den1 \leftarrow y^T s$ . (*  $den1 = y^T s$  *)
5: IF ( $den1 > (epsmaq)^{1/2} \times \|s\|_2 \times \|y\|_2$ ) THEN
    (* En caso contrario, el algoritmo termina *)
    5.1:  $den2 \leftarrow 0.0$ .
    5.2: FOR  $i = 1$  TO  $n$  DO
        5.2.1:  $u[i] \leftarrow \sum_{j=i}^n L[j,i] \times s[j]$ . (*  $u = L^T s$  *)
        5.2.2:  $den2 \leftarrow den2 + u[i]^2$ . (*  $den2 = s^T L L^T s$  *)
    5.3:  $alpha \leftarrow (den1 / den2)^{1/2}$ . (*  $\alpha = (y^T s / s^T L L^T s)^{1/2}$  *)
    5.4: IF noact THEN
        (* Sólo en la primera iteración *)
        5.4.1: FOR  $i = 1$  TO  $n$  DO
            5.4.1.1:  $u[i] \leftarrow alpha \times u[i]$ . (*  $u = \alpha \times u$  *)
            5.4.1.2: FOR  $j = i$  TO  $n$  DO
                 $L[j,i] \leftarrow alpha \times L[j,i]$ . (*  $L = \alpha \times L$  *)
        5.4.2: noact  $\leftarrow$  FALSE.

```

5.4.3: $den2 \leftarrow den1$.
5.4.4: $alpha \leftarrow 1.0$.
5.5: $reltol \leftarrow (epsmaq)^{1/2}$.
5.6: $err \leftarrow \text{TRUE}$.
5.7: FOR $i = 1$ TO n DO
 $v[i] \leftarrow \sum_{j=i}^n L[i,j] \times u[j]$. (* $v = LL^T s$ *)
5.8: $i \leftarrow 1$.
5.9: WHILE ($i \leq n$) AND (err) DO
5.9.1T: IF ($|y[i] - v[i]| \geq reltol \times \max(|gold[i]|, |gnew[i]|)$) THEN
 $err \leftarrow \text{FALSE}$
5.9.1E: ELSE
 $i \leftarrow i + 1$.
5.10: IF NOT (err) THEN
5.10.1: $v \leftarrow y - alpha \times v$. (* $v = y - \alpha LL^T s$ *)
5.10.2: $u \leftarrow alpha / den1 \times u$. (* $u = L^T s / (y^T s \times s^T LL^T s)^{1/2}$ *)
(* Copiar L^T en el triángulo superior de L *)
5.10.3: FOR $i = 2$ TO n DO
5.10.3.1: FOR $j = 1$ TO $i-1$ DO
5.10.3.1.1: $L[j,i] \leftarrow L[i,j]$.
5.10.3.1.2: $L[i,j] \leftarrow 0.0$.
(* Cálculo del factor R en la descomposición de $J^T = (L^T + uv^T) = QR$ *)
5.10.4: ACTBFGS(L, u, v, n).
(* Copiar R^T en el triángulo inferior de L *)
5.10.5: FOR $i = 2$ TO n DO
5.10.5.1: FOR $j = 1$ TO $i-1$ DO
 $L[i,j] \leftarrow L[j,i]$.
(* END IF 5.10 y 5 *)
6: RETURN. (* Fin normal algoritmo CNBFGS *)

Algoritmo CONVO

Propósito.- Comprobar si las estimaciones iniciales constituyen aproximadamente un punto estacionario (gradiente igual a cero) de la función objetivo.

Llamada.- CONVO($x_0, g_0, f_0, n, sucpsmax, retcd$).

Parámetros de entrada.- [1] $n \in \mathbb{Z}$: n° de parámetros a estimar. [2] $x_0 \in E^n$: x_0 . [3] $g_0 \in E^n$: $\nabla F(x_0)$. [4] $f_0 \in E$: $F(x_0)$.

Parámetros de salida.- [1] $sucpsmax \in \mathbb{Z}$: n° de pasos consecutivos de tamaño igual al máximo

permitido. [2] $retcd \in Z$: el algoritmo devuelve $retcd = 1$ si x_0 constituye aproximadamente un punto estacionario de la función objetivo y $retcd = 0$ en caso contrario.

Descripción.- El funcionamiento y la descripción detallada de este algoritmo pueden encontrarse en Dennis y Schnabel (1983), pp. 348-9.

Algoritmo:

- 1: $grtol \leftarrow 10^{-6}$. (* Criterio de parada estándar; pueden emplearse otros *)
- 2: $sucpmax \leftarrow 0$.
- 3T: IF $\max_{1 \leq i \leq n} (|g0[i]| \times \max(|x0[i]|, 1.0) / \max(|f0|, 1.0)) \leq grtol$ THEN
 $retcd \leftarrow 1$
- 3E: ELSE
 $retcd \leftarrow 0$.
- 4: RETURN. (* Fin normal algoritmo CONV0 *)

Algoritmo CONV

Propósito.- Comprobar si se verifica alguna de las condiciones descritas más abajo para concluir el proceso iterativo de minimización de la función objetivo [3.1.18].

Llamada.- CONV($xold, xnew, gnew, fnew, n, clpcd, itm, maxits, mxt, sucpmax, retcd$).

Parámetros de entrada.- [1] $n \in Z$: n° de parámetros a estimar. [2] $xold \in E^n$: x_i . [3] $xnew \in E^n$: x_{i+1} . [4] $gnew \in E^n$: $\nabla F(x_{i+1})$. [5] $fnew \in E$: $F(x_{i+1})$. [6] $clpcd \in Z$: código de retorno del algoritmo CLP. [7] $itm \in Z$: n° de la iteración actual (i). [8] $maxits \in Z$: máximo n° de iteraciones permitido. [9] $mxt \in \{TRUE, FALSE\}$: indicador del tamaño del paso actual devuelto por el algoritmo CLP.

Parámetros de entrada-salida.- $sucpmax \in Z$: n° de pasos consecutivos de tamaño igual al máximo permitido.

Parámetros de salida.- $retcd \in Z$: código de retorno; puede tomar uno de los siguientes valores:

- 0: no se satisface ninguno de los criterios de convergencia (el algoritmo prosigue);
- 1: norma del gradiente escalado en x_{i+1} menor que la tolerancia fijada;
- 2: distancia escalada entre x_i y x_{i+1} menor que la tolerancia fijada;
- 3: longitud de paso (calculada por el algoritmo CLP) muy pequeña;
- 4: alcanzado el máximo n° de iteraciones permitido;
- 5: alcanzado el máximo n° permitido de pasos consecutivos de tamaño igual al máximo permitido.

Descripción.- El funcionamiento y la descripción detallada de este algoritmo pueden encontrarse en

Dennis y Schnabel (1983), pp. 347-8.

Algoritmo:

1: $grtol \leftarrow 10^{-6}$. (* Criterios de parada estándar; pueden emplearse otros *)
 2: $xxtol \leftarrow 10^{-6}$.
 3: $retcd \leftarrow 0$.
 4: $max1 \leftarrow \max_{1 \leq i \leq n} (|g_{new}[i]| \times \max(|x_{new}[i]|, 1.0) / \max(|f_{new}|, 1.0))$.
 5: $max2 \leftarrow \max_{1 \leq i \leq n} (|x_{new}[i] - x_{old}[i]| / \max(|x_{new}[i]|, 1.0))$.
 6A: IF ($max1 \leq grtol$) THEN
 $retcd \leftarrow 1$
 6B: ELSE IF ($max2 \leq xxtol$) THEN
 $retcd \leftarrow 2$
 6C: ELSE IF ($clpcd = 1$) THEN
 $retcd \leftarrow 3$
 6D: ELSE IF ($itm \geq maxits$) THEN
 $retcd \leftarrow 4$
 6E: ELSE IF (mxt) THEN
 6E.1: $sucpmax \leftarrow sucpmax + 1$.
 6E.2: IF ($sucpmax = 4$) THEN (* El límite 4 es arbitrario y puede cambiarse por otro *)
 $retcd \leftarrow 5$
 6F: ELSE
 $sucpmax \leftarrow 0$.
 7: RETURN. (* Fin normal algoritmo CONV *)

Algoritmo MAXFVCN

Propósito.- Maximizar la función de verosimilitud de un proceso ARMA multivariante, mediante la minimización numérica de la función objetivo [3.1.18] con un método cuasi-Newton basado en la fórmula BFGS (vid. apartados 3.1.1 y 3.1.2).

Llamada.- MAXFVCN(PHI, IPHI, THETA, ITHETA, QQ, IQQ, mu, imu, w, a, m, p, q, g, n, delta, epsmaq, npar, xnew, MCOV, sigma2, retcd).

Parámetros de entrada.- [1] $m \in Z$: n° de series (dimensión del modelo). [2] $p \in Z$: orden AR. [3] $q \in Z$: orden MA. [4] $g \in Z$: máximo(p,q). [5] $n \in Z$: n° observaciones. [6] $IPHI(k) \in Z^{m \times m}$ ($k = 1, 2, \dots, p$) (vid. algoritmo DEFPAR). [7] $ITHETA(k) \in Z^{m \times m}$ ($k = 1, 2, \dots, q$) (vid. algoritmo DEFPAR). [8] $IQQ \in Z^{m \times m}$ (vid. algoritmo DEFPAR). [9] $imu \in Z^m$ (vid. algoritmo DEFPAR).

[10] $w \in E^{mn}$: vector de observaciones. [11] $a \in E^{mn}$: vector auxiliar. [12] $\delta \in E$: tolerancia para determinar la convergencia de las matrices $\tilde{\Sigma}_j$. [13] $epsmaq \in E$: épsilon máquina. [14] $npar \in Z$: n° de parámetros a estimar.

Parámetros de entrada-salida.- (Los parámetros [1] a [4] contienen a la entrada las estimaciones iniciales y a la salida las estimaciones finales) [1] $PHI(k) \in E^{m \times m}$ ($k = 1, 2, \dots, p$): matrices AR. [2] $THETA(k) \in E^{m \times m}$ ($k = 1, 2, \dots, q$): matrices MA. [3] $QQ \in E^{m \times m}$: matriz Q de [2.1.11]. [4] $mu \in E^m$: vector de medias. [5] $xnew \in E^{npar}$: valores iniciales (a la entrada) y finales (a la salida) de los $npar$ parámetros a estimar.

Parámetros de salida.- [1] $MCOV \in E^{npar \times npar}$: matriz de covarianzas estimadas entre los parámetros. [2] $sigma2 \in E$: estimación del parámetro σ^2 de [2.1.11] según [3.1.4]. [3] $retcd \in Z$: código de causa de terminación del algoritmo; puede ser uno de los devueltos por el algoritmo CONV o $retcd = 6$, indicando que las estimaciones iniciales no son válidas.

Almacenamiento intermedio.- [1] $xold, gnew, gold, dold \in E^{npar}$. [2] $MATG \in E^{npar \times npar}$.

Descripción (vid. apartado 3.1.2).- Partiendo de unas estimaciones iniciales, el algoritmo minimiza la función objetivo [3.1.18] utilizando un método cuasi-Newton basado en la versión factorizada de la fórmula BFGS (algoritmos CNBFGS y ACTBFGS). En cada iteración, el vector gradiente se evalúa mediante diferencias finitas centrales de la función objetivo (algoritmo CGRADD) y se calcula una longitud de paso adecuada mediante un método de "line search" (algoritmo CLP). Cuando converge, se evalúa la matriz de covarianzas entre parámetros, haciendo uso de la última actualización de la matriz de segundas derivadas y se estima el parámetro σ^2 de [2.1.11].

Algoritmo:

(* Límites estándar; pueden fijarse otros *)

1: $maxits \leftarrow 15$. (* Máximo n° de iteraciones *)

2: $maxp \leftarrow 10^3 \times \max(\|xnew\|_2, 1.0)$. (* Máximo tamaño del paso *)

3: $tolp \leftarrow 10^{-6}$. (* Mínimo tamaño relativo del paso *)

(* Evaluación de [3.1.6] y [3.1.7] en las estimaciones iniciales *)

4: $FVE(PHI, THETA, QQ, mu, w, a, m, p, q, g, n, \delta, FALSE, \Pi10, \Pi20, 1.0, tmp, fvcd, epsmaq)$.

5: IF ($fvcd \neq 0$) THEN (* Estimaciones iniciales no válidas; vid. algoritmo FVE *)

 5.1: $retcd \leftarrow 6$.

 5.2: RETURN. (* Fin anormal algoritmo MAXFVCN *)

(* Función objetivo [3.1.18] y vector gradiente en las estimaciones iniciales *)

6: $fnew \leftarrow 1.0$.

7: $CGRADD(PHI, IPHI, THETA, ITHETA, QQ, IQQ, mu, imu, w, a, m, p, q, g, n, \Pi10, \Pi20, \delta, epsmaq, gnew, npar)$.

(* Comprobar si las estimaciones iniciales constituyen un punto estacionario *)

8: CONV0(*xnew, gnew, fnew, npar, sucpxmax, retcd*).

9: IF (*retcd = 0*) THEN

(* $G_0 = I_{npar}$ *)

9.1: FOR *i = 1 TO npar DO*

9.1.1: FOR *j = 1 TO npar DO*

$MATG[i,j] \leftarrow 0.0.$

9.1.2: $MATG[i,i] \leftarrow 1.0.$

(* Bucle principal: iteración del algoritmo *)

10: WHILE (*retcd = 0*) DO

10.1: $itm \leftarrow itm + 1.$

10.2: $fold \leftarrow fnew.$

10.3: $xold \leftarrow xnew.$

10.4: $gold \leftarrow gnew.$

(* Si se desea, puede presentarse el estado del proceso: *itm, xold, fold, gold* y *MATG* *)

(* Dirección de búsqueda d_i : $G_i d_i = -\nabla F(x_i)$ *)

10.5: $dold \leftarrow -gold.$

10.6: CHOLFOR(*MATG, npar, dold*).

10.7: CHOLBACK(*MATG, npar, dold*).

(* Nueva estimación $x_{i+1} = x_i + \alpha_i d_i$; $F(x_i + \alpha_i d_i) < F(x_i)$ *)

10.8: CLP(*PHI, IPHI, THETA, ITHETA, QQ, IQQ, mu, imu, w, a, m, p, q, g, n, $\Pi10, \Pi20, \delta, epsmaq, npar, xold, fold, gold, dold, xnew, \Pi1, \Pi2, fnew, maxp, tolp, mxt, clpcd$*).

(* $\nabla F(x_{i+1})$ *)

10.9: CGRADD(*PHI, IPHI, THETA, ITHETA, QQ, IQQ, mu, imu, w, a, m, p, q, g, n, $\Pi10, \Pi20, \delta, epsmaq, xnew, gnew, npar$*).

(* Comprobar si se cumple algún criterio de convergencia *)

10.10: CONV(*xold, xnew, gnew, fnew, npar, clpcd, itm, maxits, mxt, sucpxmax, retcd*).

(* Cálculo del factor de Cholesky de G_{i+1} mediante la actualización BFGS *)

10.11: CNBFGS(*xold, gold, xnew, gnew, MATG, npar, epsmaq, itm, noact*).

(* END WHILE **10** *)

(* Estimación de σ^2 según [3.1.4] *)

11: $sigma2 \leftarrow \Pi1 / (n \times m).$

(* Estimación de la matriz de covarianzas entre parámetros; vid. nota [5] al final de la Sección 3.1 *)

12: FOR *i = 1 TO npar DO*

12.1: FOR *j = 1 TO npar DO*

$xold[j] \leftarrow 0.0.$

12.2: $xold[i] \leftarrow 1.0.$

12.3: CHOLFOR(*MATG, npar, xold*).

12.4: CHOLBACK(*MATG, npar, xold*).

12.5: FOR $j = 1$ TO $npar$ DO
 $MCOV[i,j] \leftarrow 2.0 \times fnew \times xold[j] / n.$
 13: RETURN (* Fin normal algoritmo MAXFVCN *)

Algoritmo MAXFV

Propósito.- Estimar por *MVE* o por *MVA* los parámetros de un proceso ARMA multivariante.

Llamada.- MAXFV(PHI, IPHI, THETA, ITHETA, QQ, IQQ, mu, imu, w, a, m, p, q, n, delta, sigma2, MCOV, logfv, retcd).

Parámetros de entrada.- [1] $m \in \mathbb{Z}$: n° de series (dimensión del modelo). [2] $p \in \mathbb{Z}$: orden AR. [3] $q \in \mathbb{Z}$: orden MA. [4] $n \in \mathbb{Z}$: n° observaciones. [5] $IPHI(k) \in \mathbb{Z}^{m \times m}$ ($k = 1, 2, \dots, p$): si $IPHI(k)[i,j] = 1$, el elemento (i,j) -ésimo de Φ_k es un parámetro a estimar; en caso contrario, el elemento (i,j) -ésimo de Φ_k se mantiene fijo durante la estimación. [6] $ITHETA(k) \in \mathbb{Z}^{m \times m}$ ($k = 1, 2, \dots, q$): si $ITHETA(k)[i,j] = 1$, el elemento (i,j) -ésimo de Θ_k es un parámetro a estimar; en caso contrario, el elemento (i,j) -ésimo de Θ_k se mantiene fijo durante la estimación. [7] $IQQ \in \mathbb{Z}^{m \times m}$: si $IQQ[i,j] = 1$, el elemento (i,j) -ésimo de la matriz Q es un parámetro a estimar; caso contrario, el elemento (i,j) -ésimo de Q se mantiene fijo durante la estimación. [8] $imu \in \mathbb{Z}^m$: si $imu[i] = 1$, el elemento i -ésimo de μ es un parámetro a estimar; en caso contrario, el elemento i -ésimo de μ se mantiene fijo durante la estimación. [9] $w \in E^{mn}$: vector de observaciones (series estocásticas transformadas y diferenciadas); la observación t -ésima de la i -ésima serie (elemento i -ésimo de w_t) se almacena en $w[i+(t-1)m]$ ($t = 1, 2, \dots, n$; $i = 1, 2, \dots, m$). [10] $\delta \in E$: tolerancia para determinar la convergencia de las matrices $\tilde{\Sigma}_i$ (vid. algoritmo CXI); si se desea una estimación por *MVE* debe entrar $\delta < 0$; si se desea una estimación por *MVA* debe entrar $\delta > 0$ (por ejemplo, $\delta = 10^{-3}$).

Parámetros de entrada-salida.- (Los parámetros [1] a [4] contienen a la entrada las estimaciones iniciales y a la salida las estimaciones por *MVE* o por *MVA*) [1] $PHI(k) \in E^{m \times m}$ ($k = 1, 2, \dots, p$): matrices AR; el elemento (i,j) -ésimo de Φ_k se almacena en $PHI(k)[i,j]$ ($k = 1, 2, \dots, p$; $i, j = 1, 2, \dots, m$). [2] $THETA(k) \in E^{m \times m}$ ($k = 1, 2, \dots, q$): matrices MA; el elemento (i,j) -ésimo de Θ_k se almacena en $THETA(k)[i,j]$ ($k = 1, 2, \dots, q$; $i, j = 1, 2, \dots, m$). [3] $QQ \in E^{m \times m}$: matriz Q de [2.1.11]; el elemento (i,j) -ésimo de Q se almacena en $QQ[i,j]$ ($i = 1, 2, \dots, m$; $j = i, i+1, \dots, m$; tan sólo se utiliza el triángulo superior y la diagonal principal, aunque se devuelve la matriz completa). [4] $mu \in E^n$: vector de medias; el elemento i -ésimo de μ se almacena en $mu[i]$ ($i = 1, 2, \dots, m$).

Parámetros de salida.- [1] $a \in E^{mn}$: vector de residuos evaluados en las estimaciones finales; $a[i+(t-1)m]$ contiene el elemento i -ésimo del vector \hat{a}_t ($t = 1, 2, \dots, n$; $i = 1, 2, \dots, m$). [2]

$\sigma^2 \in E$: estimación del factor σ^2 de [2.1.11]. [3] $MCOV \in E^{npar \times npar}$: matriz de covarianzas estimadas entre los parámetros. [4] $\logfv \in E$: logaritmo de la función de verosimilitud (vid. [3.1.3]) evaluada en las estimaciones finales. [5] $retcd \in Z$: código de causa de terminación del algoritmo; puede ser uno de los devueltos por el algoritmo CONV o $retcd = 6$, indicando que las estimaciones iniciales no son válidas.

Almacenamiento intermedio.- Un vector $x \in E^{npar}$, donde $npar$ es el n° de parámetros a estimar (calculado por el algoritmo).

Consideraciones adicionales.- [1] Este algoritmo sólo actúa de intermediario (*driver*) entre el usuario y el algoritmo de estimación propiamente dicho (algoritmo MAXFVCN). [2] Es recomendable que las estimaciones iniciales (incluidas las de la matriz Q) tengan un orden de magnitud homogéneo; dichas estimaciones pueden tomarse de los resultados obtenidos al estimar el modelo considerado por otros procedimientos más rápidos (*MVC*, *MSC con retrovisión* o *MSC condicional*). [3] Cuando $m = 1$ (modelos univariantes), la descomposición [2.1.11] no tiene sentido, por lo que $QQ[1,1]$ debe entrar igual a 1.0 e $IQQ[1,1]$ igual a 0 (sólo es necesario estimar σ^2). [4] El algoritmo devuelve en $MCOV$ la matriz de covarianzas estimadas entre parámetros, que puede utilizarse posteriormente para evaluar las desviaciones típicas y la matriz de correlaciones. [5] Para estimar la matriz $E[\hat{a}\hat{a}^T]$ debe realizarse posteriormente el producto de σ^2 por cada componente de QQ (vid. [2.1.11] y nota [6] al final de la Sección 3.1); a este respecto, es importante señalar que las covarianzas devueltas en $MCOV$ se refieren a los parámetros de QQ , no a los de $E[\hat{a}\hat{a}^T]$. [6] Para instrumentar en *FORTRAN*, *Pascal* o *C* el almacenamiento de las matrices Φ_k , Θ_k (y sus correspondientes *IPHI* e *ITHETA*), vid. la introducción al Apéndice A.2. [7] Si el único tipo de restricciones consideradas consiste en fijar el valor de algunos parámetros de las matrices Φ_i , Θ_i , μ y Q , el algoritmo no requiere modificaciones. Por el contrario, si se contemplan otro tipo de restricciones, éstas deben poder ser resueltas por sustitución y, en tal caso (i) debe codificarse explícitamente la dependencia de cada componente de las matrices Φ_i , Θ_i , μ y Q respecto a los parámetros que se desea estimar (vid. algoritmo DEFPAR), (ii) deben sustituirse las líneas 3-7 por la definición explícita de las variables x y $npar$ y (iii) todas las referencias a los parámetros *IPHI*, *ITHETA*, *imu* e *IQQ* deben ser suprimidas de todo el sistema.

Descripción.- En primer lugar, se define el vector de parámetros a estimar sobre la base de las especificaciones de entrada. A continuación, se estima el modelo considerado mediante una llamada al optimizador MAXFVCN. Por último, sobre la base de las estimaciones obtenidas, se evalúa el vector de residuos y el logaritmo de la función de verosimilitud mediante una última llamada al algoritmo [FVE].

Algoritmo:

(* Asignación de valores iniciales *)

1: $\logfv \leftarrow 0.0$. $retcd \leftarrow 0$. $\sigma2 \leftarrow 0.0$.

2: FOR $i = 1$ TO $m \times n$ DO $a[i] \leftarrow 0.0$.

(* Definición del vector de parámetros a estimar *)

3: $npar \leftarrow 0$.

4: FOR $i = 1$ TO m DO

4.1: IF ($imu[i] = 1$) THEN

4.1.1: $npar \leftarrow npar + 1$.

4.1.2: $x[npar] \leftarrow mu[i]$.

5: FOR $k = 1$ TO p DO

5.1: FOR $j = 1$ TO m DO

5.1.1: FOR $i = 1$ TO m DO

5.1.1.1: IF ($IPHI(k)[i,j] = 1$) THEN

5.1.1.1.1: $npar \leftarrow npar + 1$.

5.1.1.1.2: $x[npar] \leftarrow PHI(k)[i,j]$.

6: FOR $k = 1$ TO q DO

6.1: FOR $j = 1$ TO m DO

6.1.1: FOR $i = 1$ TO m DO

6.1.1.1: IF ($ITHETA(k)[i,j] = 1$) THEN

6.1.1.1.1: $npar \leftarrow npar + 1$.

6.1.1.1.2: $x[npar] \leftarrow THETA(k)[i,j]$.

7: FOR $i = 1$ TO m DO

7.1: FOR $j = i$ TO m DO

7.1.1: IF ($IQQ[i,j] = 1$) OR ($IQQ[j,i] = 1$) THEN

7.1.1.1: $npar \leftarrow npar + 1$.

7.1.1.2: $x[npar] \leftarrow QQ[i,j]$.

(* Cálculos auxiliares *)

8: **CEPSMQ**($epsmaq$).

9: $g \leftarrow \max(p,q)$.

10: FOR $i = 1$ TO $npar$ DO

10.1: FOR $j = 1$ TO $npar$ DO $MCOV[i,j] \leftarrow 0.0$.

(* Estimación de los parámetros mediante el algoritmo **MAXFVCN** *)

11: **MAXFVCN**(PHI , $IPHI$, $THETA$, $ITHETA$, QQ , IQQ , mu , imu , w , a , m , p , q , g , n , δ , $epsmaq$, $npar$, x , $MCOV$, $\sigma2$, $retcd$).

(* Evaluación del logaritmo de la función de verosimilitud y del vector de residuos *)

12: **FVE**(PHI , $THETA$, QQ , mu , w , a , m , p , q , g , n , δ , **TRUE**, $tmp1$, $tmp2$, $\sigma2$, \logfv , $intmp1$, $epsmaq$).

13: **RETURN** (* Fin normal algoritmo **MAXFV** *)

APENDICE A.4

El vector gradiente de la función objetivo

Para minimizar [3.1.8] mediante un método de propósito general (*vid.* nota [1] al final de la Sección 3.1) utilizando primeras derivadas analíticas, es necesario evaluar el vector gradiente de dicha función objetivo. Derivando [3.1.8] con respecto a un parámetro genérico x_j (que será alguna componente de Φ_k , Θ_k , Q o μ) se obtiene la j -ésima componente de su vector gradiente, que es:

$$\begin{aligned}\frac{\partial \Pi(x)}{\partial x_j} &= m(\eta^T \eta - \lambda^T \lambda)^{m-1} \times \frac{\partial (\eta^T \eta - \lambda^T \lambda)}{\partial x_j} \times |Q| \times |D|^{\frac{1}{n}} + \\ &(\eta^T \eta - \lambda^T \lambda)^m \times \frac{\partial |Q|}{\partial x_j} \times |D|^{\frac{1}{n}} + \\ &(\eta^T \eta - \lambda^T \lambda)^m \times |Q| \times \frac{1}{n} |D|^{\frac{1-n}{n}} \times \frac{\partial |D|}{\partial x_j}\end{aligned}\tag{A.3.1}$$

Si la función a minimizar es [3.1.15], para emplear un método específico para sumas de cuadrados con primeras derivadas analíticas (*vid.* nota [2] al final de la Sección 3.1), es necesario evaluar las derivadas de cada sumando con respecto a x_j . Derivando la i -ésima componente de [3.1.13] con respecto a un parámetro genérico x_j , se obtiene lo siguiente:

$$\begin{aligned}\frac{\partial \tilde{\eta}_i(x)}{\partial x_j} &= \frac{\partial \eta_i}{\partial x_j} \times |Q|^{\frac{1}{2m}} \times |D|^{\frac{1}{2mn}} + \\ &\eta_i \times \frac{1}{2m} |Q|^{\frac{1-2m}{2m}} \times \frac{\partial |Q|}{\partial x_j} \times |D|^{\frac{1}{2mn}} + \\ &\eta_i \times |Q|^{\frac{1}{2m}} \times \frac{1}{2mn} |D|^{\frac{1-2mn}{2mn}} \times \frac{\partial |D|}{\partial x_j}\end{aligned}\tag{A.3.2}$$

con $i = 1, 2, \dots, mn$. Por otro lado, derivando la i -ésima componente de [3.1.14] con respecto a x_j , se obtiene lo siguiente:

$$\begin{aligned} \frac{\partial \tilde{\lambda}_i(\mathbf{x})}{\partial x_j} &= \frac{\partial \lambda_i}{\partial x_j} \times |\mathbf{Q}|^{\frac{1}{2m}} \times |\mathbf{D}|^{\frac{1}{2mn}} + \\ &\lambda_i \times \frac{1}{2m} |\mathbf{Q}|^{\frac{1-2m}{2m}} \times \frac{\partial |\mathbf{Q}|}{\partial x_j} \times |\mathbf{D}|^{\frac{1}{2mn}} + \\ &\lambda_i \times |\mathbf{Q}|^{\frac{1}{2m}} \times \frac{1}{2mn} |\mathbf{D}|^{\frac{1-2mn}{2mn}} \times \frac{\partial |\mathbf{D}|}{\partial x_j} \end{aligned} \quad [\text{A.3.3}]$$

con $i = 1, 2, \dots, mg$. Finalmente, teniendo en cuenta en [A.3.1] que:

$$\frac{\partial (\boldsymbol{\eta}^T \boldsymbol{\eta} - \boldsymbol{\lambda}^T \boldsymbol{\lambda})}{\partial x_j} = 2 \boldsymbol{\eta}^T \frac{\partial \boldsymbol{\eta}}{\partial x_j} - 2 \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{\lambda}}{\partial x_j} \quad [\text{A.3.4}]$$

es evidente que para evaluar las derivadas [A.3.1], [A.3.2] y [A.3.3] es necesario evaluar previamente las derivadas de $|\mathbf{Q}|$, $\boldsymbol{\eta}$, $|\mathbf{D}|$ y $\boldsymbol{\lambda}$ con respecto a x_j . En primer lugar (*vid.* Apéndice A.1):

$$\frac{\partial |\mathbf{Q}|}{\partial x_j} = |\mathbf{Q}| \times \text{traza} \left[\mathbf{Q}^{-1} \frac{\partial \mathbf{Q}}{\partial x_j} \right] \quad [\text{A.3.5}]$$

En segundo lugar, teniendo en cuenta que $\boldsymbol{\eta}_k = \mathbf{R} \hat{\mathbf{a}}_{0k}$ ($k = 1, 2, \dots, n$; *vid.* [2.3.22]-[2.3.23]), es evidente que:

$$\frac{\partial \eta_k}{\partial x_j} = \frac{\partial \mathbf{R}}{\partial x_j} \hat{\mathbf{a}}_{0k} + \mathbf{R} \frac{\partial \hat{\mathbf{a}}_{0k}}{\partial x_j} \quad [\text{A.3.6}]$$

donde (*vid.* Apéndice A.1):

$$\frac{\partial \mathbf{R}}{\partial x_j} = -\frac{1}{2} \mathbf{R} \frac{\partial \mathbf{Q}}{\partial x_j} \mathbf{Q}^{-1} \quad [\text{A.3.7}]$$

y derivando [2.3.2] con respecto a x_j :

$$\frac{\partial \hat{\mathbf{a}}_{0k}}{\partial x_j} = \frac{\partial \tilde{\mathbf{w}}_k}{\partial x_j} - \sum_{i=1}^p \left[\frac{\partial \boldsymbol{\Phi}_i}{\partial x_j} \tilde{\mathbf{w}}_{k-i} + \boldsymbol{\Phi}_i \frac{\partial \tilde{\mathbf{w}}_{k-i}}{\partial x_j} \right] + \sum_{i=1}^q \left[\frac{\partial \boldsymbol{\Theta}_i}{\partial x_j} \hat{\mathbf{a}}_{0,k-i} + \boldsymbol{\Theta}_i \frac{\partial \hat{\mathbf{a}}_{0,k-i}}{\partial x_j} \right] \quad [\text{A.3.8}]$$

con $k = 1, 2, \dots, n$. Nótese que $\tilde{\mathbf{w}}_i = \hat{\mathbf{a}}_{0i} = \partial \tilde{\mathbf{w}}_i / \partial x_j = \partial \hat{\mathbf{a}}_{0i} / \partial x_j = \mathbf{0}$ para $i < 1$. En tercer lugar, para evaluar $\partial |\mathbf{D}| / \partial x_j$ es necesario evaluar previamente la derivada de la matriz \mathbf{D} con respecto a x_j , ya que (*vid.* Apéndice A.1):

$$\frac{\partial |\mathbf{D}|}{\partial x_j} = |\mathbf{D}| \times \text{traza} \left[\mathbf{D}^{-1} \frac{\partial \mathbf{D}}{\partial x_j} \right] \quad [\text{A.3.9}]$$

Entonces, teniendo en cuenta [3.1.2]:

$$\frac{\partial \mathbf{D}}{\partial x_j} = \frac{\partial \mathbf{M}^T}{\partial x_j} \mathbf{H}^T \mathbf{H} \mathbf{M} + \mathbf{M}^T \frac{\partial (\mathbf{H}^T \mathbf{H})}{\partial x_j} \mathbf{M} + \mathbf{M}^T \mathbf{H}^T \mathbf{H} \frac{\partial \mathbf{M}}{\partial x_j} \quad [\text{A.3.10}]$$

Para evaluar esta expresión, hay que tener en cuenta que (*vid.* [2.3.33] y Apéndice A.1):

$$\frac{\partial \mathbf{M}}{\partial x_j} = -\frac{1}{2} (\mathbf{V}_1 \boldsymbol{\Omega} \mathbf{V}_1^T) \frac{\partial (\mathbf{V}_1 \boldsymbol{\Omega} \mathbf{V}_1^T)^{-1}}{\partial x_j} \mathbf{M} \quad [\text{A.3.11}]$$

donde (*vid.* Apéndice A.1):

$$\frac{\partial (\mathbf{V}_1 \boldsymbol{\Omega} \mathbf{V}_1^T)^{-1}}{\partial x_j} = -(\mathbf{V}_1 \boldsymbol{\Omega} \mathbf{V}_1^T)^{-1} \frac{\partial (\mathbf{V}_1 \boldsymbol{\Omega} \mathbf{V}_1^T)}{\partial x_j} (\mathbf{V}_1 \boldsymbol{\Omega} \mathbf{V}_1^T)^{-1} \quad [\text{A.3.12}]$$

y en [A.3.12]:

$$\frac{\partial (\mathbf{V}_1 \boldsymbol{\Omega} \mathbf{V}_1^T)}{\partial x_j} = \frac{\partial \mathbf{V}_1}{\partial x_j} \boldsymbol{\Omega} \mathbf{V}_1^T + \mathbf{V}_1 \frac{\partial \boldsymbol{\Omega}}{\partial x_j} \mathbf{V}_1^T + \mathbf{V}_1 \boldsymbol{\Omega} \frac{\partial \mathbf{V}_1^T}{\partial x_j} \quad [\text{A.3.13}]$$

Dado que los elementos de \mathbf{V}_1 sólo dependen de los parámetros AR y MA (*vid.* [2.1.18], [2.1.21] y [2.1.22]), para evaluar [A.3.13] tan sólo hay que calcular la derivada $\partial \boldsymbol{\Omega} / \partial x_j$, lo cual implica (*vid.* [2.2.43]-[2.2.46]) evaluar las expresiones $\partial \Gamma_{wa}(-k) / \partial x_j$ ($k = 0, 1, \dots, q$) y $\partial \Gamma(k) / \partial x_j$ ($k = 0, 1, \dots, p$). Para el primer cálculo, es inmediato comprobar, por [2.2.6], que:

$$\frac{\partial \Gamma_{wa}(-k)}{\partial x_j} = \sum_{i=1}^k \left[\frac{\partial \Phi_i}{\partial x_j} \Gamma_{wa}(-k+i) + \Phi_i \frac{\partial \Gamma_{wa}(-k+i)}{\partial x_j} \right] - \frac{\partial \Theta_k}{\partial x_j} \mathbf{Q} - \Theta_k \frac{\partial \mathbf{Q}}{\partial x_j} \quad [\text{A.3.14}]$$

con $k = 1, 2, \dots, q$ y $\partial \Phi_i / \partial x_j = \partial \Theta_i / \partial x_j = \mathbf{0}$ para $i > p$ e $i > q$, respectivamente. Para el segundo cálculo, derivando la expresión [2.2.12] con respecto a x_j , se obtiene un sistema de $m^2(p+1)$ ecuaciones lineales con $m^2(p+1)$ incógnitas (los m^2 elementos de cada uno de los $p+1$ vectores $\partial \boldsymbol{\gamma}(k) / \partial x_j$, $k = 0, 1, \dots, p$). Dicho sistema puede escribirse de forma compacta como $\Phi \boldsymbol{\beta} = \boldsymbol{\pi}$, donde $\Phi = \mathbf{F}_1 + \mathbf{F}_2$ (*vid.* [2.2.13] y [2.2.14]), $\boldsymbol{\beta}^T = [\partial \boldsymbol{\gamma}(0)^T / \partial x_j, \partial \boldsymbol{\gamma}(1)^T / \partial x_j, \dots, \partial \boldsymbol{\gamma}(p)^T / \partial x_j]$, $\boldsymbol{\pi}^T = [\boldsymbol{\pi}(0)^T, \boldsymbol{\pi}(1)^T, \dots, \boldsymbol{\pi}(p)^T]$ y los

elementos del vector $\boldsymbol{\pi}$ se calculan como sigue:

$$\boldsymbol{\pi}(k) = \frac{\partial \omega(k)}{\partial x_j} + \sum_{i=1}^k \left[\mathbf{I}_m \otimes \frac{\partial \Phi_i}{\partial x_j} \right] \mathbf{P} \boldsymbol{\gamma}(k-i) + \sum_{i=k+1}^p \left[\mathbf{I}_m \otimes \frac{\partial \Phi_i}{\partial x_j} \right] \boldsymbol{\gamma}(i-k) \quad [\text{A.3.15}]$$

para $k = 0, 1, \dots, p$, donde $\partial \omega(k)/\partial x_j$ puede obtenerse aplicando el operador *vec* a la siguiente expresión (vid. [2.2.10]):

$$\frac{\partial \mathbf{W}(k)}{\partial x_j} = \frac{\partial \boldsymbol{\Gamma}_{wa}(k)^T}{\partial x_j} - \sum_{i=1}^q \left[\frac{\partial \boldsymbol{\Theta}_i}{\partial x_j} \boldsymbol{\Gamma}_{wa}(k-i)^T + \boldsymbol{\Theta}_i \frac{\partial \boldsymbol{\Gamma}_{wa}(k-i)^T}{\partial x_j} \right] \quad [\text{A.3.16}]$$

Por tanto, una vez que se dispone de todas las cantidades necesarias para evaluar [A.3.11], tan sólo hay que calcular $\partial(\mathbf{H}^T \mathbf{H})/\partial x_j$ para poder evaluar [A.3.10]. Si definimos $\mathbf{P} = \mathbf{H}^T \mathbf{H}$, teniendo en cuenta [2.3.9], [2.3.10] y [2.2.22], es inmediato comprobar que, para $i = 1, 2, \dots, g$:

$$\frac{\partial \mathbf{P}_{i1}}{\partial x_j} = \sum_{k=0}^{n-i} \left[\frac{\partial \boldsymbol{\Xi}_k^T}{\partial x_j} \mathbf{Q}^{-1} \boldsymbol{\Xi}_{k+i-1} + \boldsymbol{\Xi}_k^T \frac{\partial \mathbf{Q}^{-1}}{\partial x_j} \boldsymbol{\Xi}_{k+i-1} + \boldsymbol{\Xi}_k^T \mathbf{Q}^{-1} \frac{\partial \boldsymbol{\Xi}_{k+i-1}}{\partial x_j} \right] \quad [\text{A.3.17}]$$

y para $i = 2, 3, \dots, g$, $h = 2, 3, \dots, i$:

$$\frac{\partial \mathbf{P}_{ih}}{\partial x_j} = \frac{\partial \mathbf{P}_{i-1, h-1}}{\partial x_j} - \frac{\partial \boldsymbol{\Xi}_{n-i+1}^T}{\partial x_j} \mathbf{Q}^{-1} \boldsymbol{\Xi}_{n-h+1} - \boldsymbol{\Xi}_{n-i+1}^T \frac{\partial \mathbf{Q}^{-1}}{\partial x_j} \boldsymbol{\Xi}_{n-h+1} - \boldsymbol{\Xi}_{n-i+1}^T \mathbf{Q}^{-1} \frac{\partial \boldsymbol{\Xi}_{n-h+1}}{\partial x_j} \quad [\text{A.3.18}]$$

Finalmente, para evaluar [A.3.17] y [A.3.18] hay que tener en cuenta que, por [2.3.8]:

$$\frac{\partial \boldsymbol{\Xi}_i}{\partial x_j} = \sum_{k=1}^q \left[\frac{\partial \boldsymbol{\Theta}_k}{\partial x_j} \boldsymbol{\Xi}_{i-k} + \boldsymbol{\Theta}_k \frac{\partial \boldsymbol{\Xi}_{i-k}}{\partial x_j} \right] \quad [\text{A.3.19}]$$

y, por otro lado (vid. Apéndice A.1):

$$\frac{\partial \mathbf{Q}^{-1}}{\partial x_j} = -\mathbf{Q}^{-1} \frac{\partial \mathbf{Q}}{\partial x_j} \mathbf{Q}^{-1} \quad [\text{A.3.20}]$$

En definitiva, las expresiones [A.3.10]-[A.3.20] permiten evaluar [A.3.9], por lo que para evaluar las derivadas [A.3.1]-[A.3.3] de la función objetivo tan sólo resta obtener la expresión analítica de $\partial \lambda / \partial x_j$. Para ello, tan sólo hay que tener en cuenta que, como $\boldsymbol{\lambda}$ es la solución del sistema $\mathbf{L} \boldsymbol{\lambda} = \mathbf{M}^T \tilde{\mathbf{h}}$,

entonces $\lambda = \mathbf{L}^{-1} \mathbf{M}^T \tilde{\mathbf{h}}$, por lo que:

$$\frac{\partial \lambda}{\partial x_j} = \frac{\partial \mathbf{L}^{-1}}{\partial x_j} \mathbf{M}^T \tilde{\mathbf{h}} + \mathbf{L}^{-1} \frac{\partial \mathbf{M}^T}{\partial x_j} \tilde{\mathbf{h}} + \mathbf{L}^{-1} \mathbf{M}^T \frac{\partial \tilde{\mathbf{h}}}{\partial x_j} \quad [\text{A.3.21}]$$

donde, teniendo en cuenta [A.3.11], tan sólo hay que evaluar (*vid.* [2.3.19]):

$$\frac{\partial \mathbf{h}_k}{\partial x_j} = \sum_{i=0}^{n-k} \left[\frac{\partial \tilde{\mathbf{x}}_i^T}{\partial x_j} \mathbf{R}^T \eta_{i+k} + \tilde{\mathbf{x}}_i^T \frac{\partial \mathbf{R}^T}{\partial x_j} \eta_{i+k} + \tilde{\mathbf{x}}_i^T \mathbf{R}^T \frac{\partial \eta_{i+k}}{\partial x_j} \right] \quad [\text{A.3.22}]$$

con $k = 1, 2, \dots, g$, y (*vid.* Apéndice A.1):

$$\frac{\partial \mathbf{L}^{-1}}{\partial x_j} = -\mathbf{L}^{-1} \frac{\partial \mathbf{L}}{\partial x_j} \mathbf{L}^{-1} \quad [\text{A.3.23}]$$

Todas las derivadas que aparecen en [A.3.22] ya están calculadas (*vid.* [A.3.6]-[A.3.8] y [A.3.19]), mientras que para evaluar [A.3.23] tan sólo hay que tener en cuenta que, como \mathbf{L} es el factor de Cholesky de la matriz \mathbf{D} de [3.1.2], entonces (*vid.* Apéndice A.1):

$$\frac{\partial \mathbf{L}}{\partial x_j} = -\frac{1}{2} \mathbf{D} \frac{\partial \mathbf{D}^{-1}}{\partial x_j} \mathbf{L} \quad [\text{A.3.24}]$$

donde:

$$\frac{\partial \mathbf{D}^{-1}}{\partial x_j} = -\mathbf{D}^{-1} \frac{\partial \mathbf{D}}{\partial x_j} \mathbf{D}^{-1} \quad [\text{A.3.25}]$$

y esta última expresión puede evaluarse haciendo uso de [A.3.10].

En resumen, el cálculo ordenado de las derivadas que aparecen en [A.3.1]-[A.3.3] puede llevarse a cabo mediante el siguiente procedimiento (algoritmo **[AGRD]**):

[AGRD.1] Cálculo de $\partial |\mathbf{Q}| / \partial x_j$ (evaluación directa de [A.3.5]).

[AGRD.2] Cálculo de $\partial \eta / \partial x_j$:

[2.1] Evaluar [A.3.8] con $k = 1, 2, \dots, n$.

[2.2] Evaluar [A.3.7].

[2.3] Evaluar [A.3.6] con $k = 1, 2, \dots, n$.

[AGRD.3] Cálculo de $\partial |D| / \partial x_j$:

[3.1] Evaluar [A.3.14] con $k = 1, 2, \dots, q$.

[3.2] Evaluar [A.3.16] y [A.3.15] con $k = 0, 2, \dots, p$.

[3.3] Resolver para β el sistema $\Phi\beta = \pi$ con $\Phi = F_1 + F_2$ (vid. [2.2.3] y [2.2.4]) y $\pi^T = [\pi(0)^T, \pi(1)^T, \dots, \pi(p)^T]$, donde los $\pi(k)$ son el resultado del paso [3.2].

[3.4] Formar la matriz $\partial\Omega/\partial x_j$ a partir de los pasos [3.1] y [3.3] (vid. [2.2.43]-[2.2.46]).

[3.5] Evaluar, por este orden, [A.3.13], [A.3.12] y [A.3.11] ($\partial M/\partial x_j$).

[3.6] Evaluar [A.3.20] y [A.3.19].

[3.7] Evaluar [A.3.17] y [A.3.18] [$\partial(H^T H)/\partial x_j$].

[3.8] Evaluar [A.3.10] ($\partial D/\partial x_j$) haciendo uso de [3.5] y [3.7].

[3.9] Evaluar [A.3.9].

[AGRD.4] Cálculo de $\partial\lambda/\partial x_j$:

[4.1] Evaluar [A.3.22] con $k = 1, 2, \dots, g$ haciendo uso de cantidades evaluadas en [AGR.2] y [3.6].

[4.2] Evaluar [A.3.25] haciendo uso de [3.8].

[4.3] Evaluar, por este orden, [A.3.24] y [A.3.23].

[4.4] Evaluar [A.3.21].

Para concluir, es importante señalar que el procedimiento [AGRD] ha de llevarse a cabo, en cada iteración del proceso de minimización de [3.1.8] o de [3.1.15], un número de veces igual al número de parámetros a estimar. La elevada carga computacional implícita en esa tarea sugiere la conveniencia de evaluar numéricamente las primeras derivadas de la función objetivo. No obstante, podría tener interés codificar el procedimiento [AGRD] y hacer uso de él en la estimación, aunque a partir de la experiencia práctica resumida en los resultados que se presentan en el Capítulo 3, esa tarea no se ha considerado fundamental.

APENDICE A.5

Simulación de procesos ARMA multivariantes

En este Apéndice se describe un procedimiento para simular observaciones generadas por un proceso ARMA(p, q) multivariante estocástico. Las series obtenidas con este procedimiento se han empleado para realizar los ejercicios de estimación, en situaciones simuladas, que se presentan en el Capítulo 3. Básicamente, el procedimiento utilizado es una simple extensión al contexto multivariante de la metodología descrita en Ansley y Newbold (1980, pp. 181-2).

El problema consiste en obtener n observaciones m -variantes \mathbf{w}_t ($t=1, 2, \dots, n$) generadas recursivamente por la siguiente ecuación (*vid.* [2.1.3]-[2.1.5]):

$$\mathbf{w}_t = \boldsymbol{\mu} + \sum_{i=1}^p \boldsymbol{\Phi}_i \tilde{\mathbf{w}}_{t-i} + \mathbf{a}_t - \sum_{i=1}^q \boldsymbol{\Theta}_i \mathbf{a}_{t-i} \quad [\text{A.5.1}]$$

donde $\tilde{\mathbf{w}}_t = \mathbf{w}_t - \boldsymbol{\mu}$ y \mathbf{a}_t ($t=1, 2, \dots, n$) es una sucesión de n observaciones independientes generadas por una distribución Normal m -variante, caracterizada por:

$$E[\mathbf{a}_t] = \mathbf{0} \quad [\text{A.5.2}]$$

$$E[\mathbf{a}_t \mathbf{a}_{t+k}^T] = \begin{cases} \sigma^2 \mathbf{Q}_{(m \times m)}, & k=0 \\ \mathbf{0}_{(m \times m)}, & k \neq 0 \end{cases} \quad [\text{A.5.3}]$$

con $\sigma^2 > 0$ y \mathbf{Q} simétrica y definida positiva. En [A.5.1]-[A.5.3], $\boldsymbol{\mu}$, $\boldsymbol{\Phi}_i$ ($i=1, 2, \dots, p$), $\boldsymbol{\Theta}_i$ ($i=1, 2, \dots, q$), σ^2 y \mathbf{Q} son parámetros conocidos.

Para evaluar [A.5.1] desde $t=1$ hasta $t=n$, es necesario disponer de los $p+q$ valores iniciales $\tilde{\mathbf{w}}_{1-p}, \tilde{\mathbf{w}}_{2-p}, \dots, \tilde{\mathbf{w}}_0, \mathbf{a}_{1-q}, \mathbf{a}_{2-q}, \dots, \mathbf{a}_0$, que recogemos en el vector \mathbf{u}_* (*vid.* [2.1.16]). Es inmediato comprobar que $\mathbf{u}_* \sim N(\mathbf{0}, \sigma^2 \boldsymbol{\Omega})$, donde $\boldsymbol{\Omega}$ es la matriz de [2.2.43]-[2.2.47]. Entonces, para estimar el vector \mathbf{u}_* , tan sólo se necesita disponer de una sucesión \mathbf{e}_t ($t=1, 2, \dots, p+q$) de $p+q$ observaciones independientes generadas por una distribución Normal m -variante, caracterizada por:

$$E[\mathbf{e}_t] = \mathbf{0}_{(m \times 1)} \quad [\text{A.5.4}]$$

$$E[\mathbf{e}_t \mathbf{e}_{t+k}^T] = \begin{cases} \sigma^2 \mathbf{I}_{(m \times m)}, & k=0 \\ \mathbf{0}_{(m \times m)}, & k \neq 0 \end{cases} \quad [\text{A.5.5}]$$

ya que, si definimos $\hat{\mathbf{u}}_* = \mathbf{\Omega}_1 \mathbf{e}$, donde $\mathbf{e}^T = [\mathbf{e}_1^T, \mathbf{e}_2^T, \dots, \mathbf{e}_{p+q}^T]$ y $\mathbf{\Omega}_1$ es el factor de Cholesky de $\mathbf{\Omega}$ ($\mathbf{\Omega} = \mathbf{\Omega}_1 \mathbf{\Omega}_1^T$), entonces es evidente que $\hat{\mathbf{u}}_*$ tiene la distribución apropiada para constituir un conjunto adecuado de valores iniciales (en concreto, $E[\hat{\mathbf{u}}_*] = \mathbf{0}$ y $E[\hat{\mathbf{u}}_* \hat{\mathbf{u}}_*^T] = \sigma^2 \mathbf{\Omega}$).

Sobre la base de estas consideraciones, el procedimiento que se ha utilizado para generar una realización de tamaño n de un proceso ARMA(p, q) multivariante, ha sido el siguiente:

- [1] Generar 2000 observaciones procedentes de una distribución Normal m -variante con media $\mathbf{0}$ y matriz de covarianzas $\sigma^2 \mathbf{Q}$ (vid. [A.5.2]-[A.5.3]).
- [2] Seleccionar las n últimas observaciones, que componen la sucesión \mathbf{a}_t ($t=1, 2, \dots, n$).
- [3] Generar 2000 observaciones procedentes de una distribución Normal m -variante con media $\mathbf{0}$ y matriz de covarianzas $\sigma^2 \mathbf{I}$ (vid. [A.5.4]-[A.5.5]).
- [4] Seleccionar las $p+q$ últimas observaciones, que componen la sucesión \mathbf{e}_t ($t=1, 2, \dots, p+q$).
- [5] Evaluar los elementos de la matriz $\mathbf{\Omega}$ (vid. [2.2.43]-[2.2.47]) y obtener su descomposición de Cholesky ($\mathbf{\Omega} = \mathbf{\Omega}_1 \mathbf{\Omega}_1^T$).
- [6] Evaluar los elementos del vector $\hat{\mathbf{u}}_* = \mathbf{\Omega}_1 \mathbf{e}$ y utilizarlos como valores iniciales (vid. [2.1.16]) para evaluar [A.5.1] desde $t = 1$ hasta $t = n$.

La generación de números aleatorios, contenida en los pasos [1] y [3], se ha llevado a cabo utilizando las rutinas ofrecidas por *The S.C.A. Statistical System* (vid. Hudak y Liu (1991), cap. 12), basadas en el método de Box y Muller (1958). Por último, cada una de las 100 realizaciones de \mathbf{a}_t y \mathbf{e}_t ha sido generada utilizando una *semilla* distinta en cada caso, por lo que los resultados de cada estimación pueden considerarse razonablemente independientes entre sí.

Referencias bibliográficas

Aigner, D.J. [1971]: A compendium on estimation of the autoregressive-moving average model from time series data. *International Economic Review*, 12, 3 (Oct.), pp. 348-71.

Ali, M.M. [1977]: Analysis of autoregressive moving-average models: Estimation and prediction. *Biometrika* 64, 3, pp. 535-45.

Anderson, B.D.O.; J.B. Moore [1979]: *Optimal Filtering*. Prentice-Hall, New Jersey.

Ansley, C.F. [1979]: An algorithm for the exact likelihood of a mixed autoregressive-moving average process. *Biometrika* 66, 1, pp. 59-65.

Ansley, C.F. [1980]: Computation of the theoretical autocovariances for a vector ARMA process. *Journal of Statistical Computation and Simulation*, 12, pp. 15-24.

Ansley, C.F.; R. Kohn [1983]: Exact likelihood of vector autoregressive-moving average process with missing or aggregated data. *Biometrika* 70, 1, pp. 275-8.

Ansley, C.F.; P. Newbold [1980]: Finite sample properties of estimators for autoregressive moving average models. *Journal of Econometrics* 13, pp. 159-83.

Box, G.E.P.; D.R. Cox [1964]: An analysis of transformations. *Journal of the Royal Statistical Society, Ser. B*, 26, pp. 211-52.

Box, G.E.P.; M.E. Muller [1958]: A note on the generation of random normal deviates. *Annals of Mathematical Statistics*, 29, pp. 610-11.

Box, G.E.P.; G.M. Jenkins [1970]: *Time series analysis, forecasting and control*. Holden Day, San Francisco (Edición revisada: 1976).

Brent, R.P. [1973]: *Algorithms for minimization without derivatives*. Prentice-Hall, New Jersey.

Cancio, R. [1989]: *Estimación maximoverosímil y predicción de modelos econométricos lineales dinámicos en espacio de los estados*. Tesis Doctoral, Universidad Complutense de Madrid.

Dennis, J.E.; R.B. Schnabel [1983]: *Numerical methods for unconstrained optimization and nonlinear equations*. Prentice-Hall, New Jersey.

Dent, W.T. [1977]: Computation of the exact likelihood function of an ARIMA process. *Journal of Statistical Computation and Simulation* 5, pp. 193-206.

Dent, W.T.; A.S. Min [1978]: A Monte Carlo study of autoregressive integrated moving-average processes. *Journal of Econometrics* 7, pp. 23-55.

Dhrymes, P.J. [1984]: *Econometría*. Editorial AC, Madrid.

Engle, R.F. [1983]: Wald, likelihood ratio and Lagrange multiplier tests in Econometrics. En **Griliches, Z.; M.D. Intriligator** (eds.): *Handbook of Econometrics*, vol. II. North-Holland, Amsterdam.

Flores, R. [1987]: *Análisis econométrico sobre la incidencia de la economía de Estados Unidos y economía mundial en la economía española*. Tesis Doctoral, Universidad Complutense de Madrid.

Gardner, G.; A.C. Harvey; G.D.A. Phillips [1980]: An algorithm for exact maximum likelihood estimation of autoregressive-moving average models by means of Kalman filtering. *Applied Statistics (Statistical Algorithms AS 154)* 29, pp. 311-22.

Gill, P.E.; W. Murray; M.H. Wright [1980]: *Practical optimization*. Academic Press, London.

Godolphin, E.J.; J.M. Unwin [1983]: Evaluation of the covariance matrix for the maximum likelihood estimator of a Gaussian autoregressive-moving average process. *Biometrika* 70, 1, pp. 279-84.

Granger, C.W.J.; P. Newbold [1986]: *Forecasting economic time series (2nd edition)*. Academic Press, London.

Hall, A.D.; D.F. Nicholls [1980]: The evaluation of exact maximum likelihood estimates for VARMA models. *Journal of Statistical Computation and Simulation*, 10, pp. 251-62.

Hannan, E.J. [1970]: *Multiple time series*. John Wiley, New York.

Hillmer, S.C.; G.C. Tiao [1979]: Likelihood function of stationary multiple autoregressive moving average models. *Journal of the American Statistical Association* 74 (367), pp. 652-60.

Hudak, G.B.; L.M. Liu [1991]: *The S.C.A. Statistical System. Reference manual for general statistical analysis*. Scientific Computing Associates, Dekalb.

Jenkins, G.M.; A.S. Alavi [1981]: Some aspects of modelling and forecasting multivariate time series. *Journal of Time Series Analysis* 2, 1, pp. 1-47.

Jones, R.H. [1980]: Maximum likelihood fitting of ARMA models to time series with missing observations. *Technometrics* 22, pp. 389-95.

Kabaila, P. [1980]: An optimality property of the least-squares estimate of the parameter of the spectrum of a purely nondeterministic time series. *Annals of Statistics*, 8, pp. 1082-92.

Kabaila, P. [1983]: On the asymptotic efficiency of estimators of the parameters of an ARMA process. *Journal of Time Series Analysis*, 4, 1, pp. 37-47.

Kalman, R.E. [1960]: A new approach to linear filtering and prediction problems. *Journal of Basic Engineering Trans. ASME*, 82-D, pp. 33-45.

Kohn, R.; C.F. Ansley [1982]: A note on obtaining the theoretical autocovariances of an ARMA process. *Journal of Statistical Computation and Simulation* 15, pp. 273-83.

Kohn, R.; C.F. Ansley [1985]: Computing the likelihood and its derivatives for a Gaussian ARMA model. *Journal of Statistical Computation and Simulation* 22, pp. 229-63.

Koreisha, S.; T. Pukkila [1989]: Fast linear estimation methods for vector autoregressive moving-average models. *Journal of Time Series Analysis*, 10, 4, pp. 325-39.

Koreisha, S.; T. Pukkila [1990]: A generalized least-squares approach for estimation of autoregressive moving-average models. *Journal of Time Series Analysis*, 11, 2, pp. 139-51.

- Koreisha, S.; T. Pukkila [1990]:** A generalized least-squares approach for estimation of autoregressive moving-average models. *Journal of Time Series Analysis*, 11, 2, pp. 139-51.
- Liu, L.M.; G.B. Hudak [1986]:** *The S.C.A. Statistical System. Reference manual for forecasting and time series analysis*. Scientific Computing Associates, DeKalb.
- Liu, L.M.; G.B. Hudak [1992]:** *Forecasting and time series analysis using the S.C.A. Statistical System (vol. I)*. Scientific Computing Associates, DeKalb.
- Ljung, G.M.; G.E.P. Box [1979]:** The likelihood function of stationary autoregressive-moving average models. *Biometrika*, 66, 2, pp. 265-70.
- Maddala, G.S. [1977]:** *Econometrics*. McGraw-Hill, New York.
- Martin, R.S.; J.H. Wilkinson [1965]:** Symmetric decomposition of positive definite band matrices. *Numerische Mathematik* 7, pp. 355-61.
- Martin, R.S.; G. Peters; J.H. Wilkinson [1965]:** Symmetric decomposition of a positive definite matrix. *Numerische Mathematik* 7, pp. 362-83.
- McLeod, A.I. [1975, 1977]:** Derivation of the theoretical autocovariance function of autoregressive moving average time series. *Applied Statistics* 24, pp. 255-56; corrección 26, pp. 194.
- Melard, G. [1984]:** A fast algorithm for the exact likelihood of autoregressive moving average models. *Applied Statistics (Statistical Algorithms AS 197)*, 33, pp. 104-14.
- Mills, T.C [1990]:** *Time series techniques for economists*. Cambridge University Press, Cambridge.
- Morf, M.; G.S. Sidhu; T. Kailath [1974]:** Some new algorithms for recursive estimation of constant, linear, discrete-time systems. *IEEE Trans. Auto. Control*, AC-19, pp. 315-23.
- Newbold, P. [1974]:** The exact likelihood function for a mixed autoregressive-moving average process. *Biometrika* 61, 3, pp. 423-26.

Nicholls, D.F. [1977]: A comparison of estimation methods for vector linear time series models. *Biometrika*, 64, 1, pp. 85-90.

Nicholls, D.F.; A.D. Hall [1979]: The exact likelihood function of multivariate autoregressive-moving average models. *Biometrika* 66, 2, pp. 259-64.

Numerical Algorithms Group [1987]: *Library Manual Mark 12*. NAG Central Office, Oxford.

Osborn, D.R. [1977]: Exact and approximate maximum likelihood estimators for vector moving average processes. *Journal of the Royal Statistical Society, Ser. B*, 39, pp. 114-18.

Pagano, M. [1973]: When is an autoregressive scheme stationary? *Communications in Statistics 1*, pp. 533-44.

Pearlman, J.G. [1980]: An algorithm for the exact likelihood of a high-order autoregressive-moving average process. *Biometrika* 67, 1, pp. 232-33.

Peña, D. [1989]: *Estadística. Modelos y métodos (2. Modelos lineales y series temporales) (2ª edición)*. Alianza Editorial, S.A., Madrid.

Phadke, M.S.; G. Kedem [1978]: Computation of the exact likelihood function of multivariate moving average models. *Biometrika* 65, 3, pp. 511-19.

Shea, B.L. [1984]: Maximum likelihood estimation of multivariate ARMA processes via the Kalman filter. En **Anderson, O.D.** (ed.): *Time series analysis: Theory and practice 5*. North Holland, Amsterdam.

Shea, B.L. [1987]: Estimation of multivariate time series. *Journal of Time Series Analysis* 8, 1, pp. 95-109.

Shea, B.L. [1989]: The exact likelihood of a vector autoregressive moving average model. *Applied Statistics (Statistical Algorithms AS 242)* 38, 1, pp. 161-204.

Sotoca, S. [1992]: *Aplicaciones econométricas del filtro de Kalman y algunas variaciones numéricas:*

el filtro de Chandrasekhar. Tesis Doctoral, Universidad Complutense de Madrid.

Tiao, G.C.; G.E.P. Box [1981]: Modeling multiple time series with applications. *Journal of the American Statistical Association* 76 (376), pp. 802-16.

Tiao, G.C.; R.S. Tsay [1983]: Multiple time series modeling and extended sample cross-correlations. *Journal of Business & Economic Statistics* 1, 1, pp. 43-56.

Tiao, G.C.; R.S. Tsay [1989]: Model specification in multivariate time series. *Journal of the Royal Statistical Society (Ser. B)*, 51, 2, pp. 157-213.

Tuan, P.D. [1987]: Exact maximum likelihood estimate and Lagrange multiplier test statistic for ARMA models. *Journal of Time Series Analysis* 8, 1, pp. 61-78.

Tunncliffe Wilson, G. [1973]: The estimation of parameters in multivariate time series models. *Journal of the Royal Statistical Society, Ser. B*, 35, pp. 76-85.

Tunncliffe Wilson, G. [1979]: Some efficient computational procedures for high order ARMA models. *Journal of Statistical Computation and Simulation*, 8, pp. 301-9.

Wallis, K.F. [1977]: Multiple time series analysis and the final form of econometric models. *Econometrica* 45, 6, pp. 1481-97.

Wei, W.W.S. [1990]: *Time series analysis (Univariate and multivariate methods)*. Addison-Wesley, Redwood City.

Whittle, P. [1953]: Estimation and information in stationary time series. *Ark. Mat. Fys. Astr.*, 2, pp. 423-34.

Zellner, A.; F. Palm [1974]: Time series analysis and simultaneous equation econometric models. *Journal of Econometrics* 2, pp. 17-54.

