Instituto Nacional de Estadística

Working Papers

05/2011

Multivariate Wiener-Kolmogorov Filtering by Polynomial Methods

Félix Aparicio-Pérez

The views expressed in this working paper are those of the authors and do not necessarily reflect the views of the Instituto Nacional de Estadística of Spain

First draft: July 2011

This draft: July 2011

Multivariate Wiener-Kolmogorov Filtering by Polynomial Methods

Abstract

The exact computation of a general multivariate Wiener-Kolmogorov filter is usually done in state-space form. This paper develops a new method to solve the problem within the polynomial framework. To do so, several matrix polynomial techniques are used. To obtain the initial values, some new techniques are also developed. Finally, some extensions and applications are outlined.

Keywords

Wiener-Kolgomorov filter, polynomial matrices

Authors and Affiliations

Félix Aparicio-Pérez

Dirección General de Metodología, Calidad y Tecnologías de la Información y las Comunicaciones, INE

Multivariate Wiener-Kolmogorov Filtering by Polynomial Methods

Félix Aparicio-Pérez, Instituto Nacional de Estadística, Madrid, Spain. [1]

Abstract The exact computation of a general multivariate Wiener-Kolmogorov filter is usually done in state-space form. This paper develops a new method to solve the problem within the polynomial framework. To do so, several matrix polynomial techniques have to be used, like the transformation of a right matrix fraction into a left one or the spectral factorization of a moving average. To obtain the initial values, some new techniques are developed, these include an extended innovations algorithm or a new kind of matrix geometric series. Finally, some extensions and applications are outlined.

Keywords: Wiener-Kolmogorov filter, polynomial matrices.

MSC2010 subject classification: 37M10

Contents

[1] e-mail: fapape@ine.es

1 Introduction

Wiener-Kolmogorov filtering dates back from the independent works of Wiener and Kolmogorov in the first half of the twentieth century, see Wiener(1949) and Kolmogorov (1941). The difficulties that were found to apply the exact filter to a finite sample, or even the semi-infinite filter to multivariate problems, the non-recursive nature of the filter and the fact that some problems need a model with time-varying coefficients motivated a growing use of the Kalman filter (Kalman, (1960)) and smoother instead of the Wiener-Kolmogorov filter. This situation is still present in our days, and it is not satisfactory, since the Kalman filter has also some drawbacks. For example, when working with a high dimensional polynomial model, there is no numerically stable way to express it in state space form: the methods that are used to transform a polynomial model into state space form work well for small dimensions, but break down when the dimension of the state is high (see, e.g., Kenney and Laub (1988)). Another problem, specific to signal extraction in time series analysis, is the fact that the Kalman filter fails to provide intuition about the filter to the researcher. In fact, most researchers prefer polynomial filtering formulae, since they are considered to be more revealing.

The main development in this paper is an implementation in the general multivariate case of the Wiener-Kolmogorov filter using polynomial techniques, but it also contains some more contributions, like an extension of the innovations algorithm to estimate random variables whose cross-correlations with a time series are known or a new method to compute the initial values of a filter. In Aparicio-Perez (2010) a brief explanation of the method was given, but the details were only available in unpublished manuscripts.

All the techniques proposed in the paper can be implemented in a numerically efficient and reliable way, so resulting in a serious competitor of state-space techniques. Finally, some extensions and applications in the field of official statistics are outlined.

The paper makes an extensive use of some matrix polynomial techniques, like the transformation of a right matrix fraction description into a left one or the computation of the inverse of a polynomial matrix. A detailed treatment of matrix polynomial techniques can be found, for example, in Kailath, T. (1980), Chen (1984) and chapters 7 and 14 in Lancaster and Tismenetsky (1985). Some specific polynomial techniques that are used in this paper are new and cannot be found in these references, but they can be found in Aparicio-Perez (2011); these include the computation of the model that follows a filtered VARMA process or the time reversion of a VARMA process.

In this paper s stands for time t reversed. For example, if we have a sample of size T of a process y_t for time points $t \in \{1, \cdots, T\}$, that we will call $y_{1:T}$, for each t we could define $s = T + 1 - t$. The time reversed process is represented by $\breve{y}_s$ and a circular accent $\breve{}$ will be also used with the elements of its VARMA model, The notation $\hat{Z}_{t_1/t_2}$ is used to denote the conditional expectation of the Z_t process at time point t_1 based on the information available up to time point t_2 or, equivalently, the projection of Z_{t_1} on the Hilbert space spanned by the observed process up to time point t_2. Sometimes the abbreviation $\hat{Z}_t$ is used, its meaning is $\hat{Z}_{t/t-1}$ if $t \leq T$ and $\hat{Z}_{t/T}$ if $t > T$. Given a polynomial matrix $P(z)$, we sometimes consider it as the sum $P(z) = P_0 + zP_1 + z^2P_2 + \cdots + z^gP_g$, where the P_i are scalar matrices and $g \triangleq \partial(P(z))$ is the degree of the polynomial matrix. Finally, F stands for the lead or forward shift operator and B stands for the lag or backward shift operator, that is $Fy_t \triangleq y_{t+1}$ and $By_t \triangleq y_{t-1}$.

2 Filter Equations

Assume that two multivariate processes, s_t and y_t, of dimensions s and r respectively, follow jointly a regular, gaussian, stationary, invertible and left coprime VARMA model $a(B)x_t = b(B)\epsilon_t$, with covariance matrix $\Sigma_\epsilon = cov(\epsilon_t)$, where $x_t' = (s_t', y_t')'$. We consider this model partitioned as in

$$\begin{pmatrix} a_{11}(B) & a_{12}(B) \\ a_{21}(B) & a_{22}(B) \end{pmatrix} \begin{pmatrix} s_t \\ y_t \end{pmatrix} = \begin{pmatrix} b_{11}(B) & b_{12}(B) \\ b_{21}(B) & b_{22}(B) \end{pmatrix} \begin{pmatrix} \epsilon_{1t} \\ \epsilon_{2t} \end{pmatrix} \tag{1}$$

We will also assume that $y_{1:T}$, a finite sample of length T of y_t is available, but that no observations from s_t are available. The main focus of this paper is on non-causal filtering, that is, on the computation of $\hat{s}_{t/T} \triangleq E[s_t/y_{1:T}]$. This is also called a fixed interval smoothing problem, and it can be solved exactly by the Kalman filter and smoother. Causal filtering, on the other hand, is used to compute $\hat{s}_{t/t-h}$ for some $h \geq 0$.

In Burman (1980), a method for the exact computation of the non-causal univariate Wiener-Kolmogorov filter based on a finite sample is explained. Some particular cases of the causal multivariate Wiener-Kolmogorv filter, like the signal plus noise problem or the deconvolution problem, have been addressed in the literature before using polynomial methods, see e.g. Ahlén and Sternad (1991).

We will develop a new polynomial method that realizes the general multivariate causal and non-causal Wiener-Kolmogorov filters.

Define the marginal model of y_t as the VARMA model that follows the filtered process $y_t = (0_{r\times s}, I_r)x_t$, where $0_{r\times s}$ is the $r \times s$ zero matrix and I_r is the identity matrix of dimension r. Let this marginal model be

$$\Phi(B)y_t = \Theta(B)u_t, \tag{2}$$

where u_t is a white noise process with $cov(u_t) = \Sigma_u$. The computation of the marginal model of y_t is a particular case of the computation of the model that follows a filtered process and can be done as explained in Aparicio-Perez (2011).

In the polynomial setting, we first transform the model (1) into another one that has a diagonal AR part. This can be accomplished by pre-multiplying (1) by $Adj(a(B))$, the adjoint of the polynomial matrix $a(B)$. The result is

$$det(a(B))I_n \begin{pmatrix} s_t \\ y_t \end{pmatrix} = \begin{pmatrix} d_{11}(B) & d_{12}(B) \\ d_{21}(B) & d_{22}(B) \end{pmatrix} \begin{pmatrix} \hat{\epsilon}_{1t} \\ \hat{\epsilon}_{2t} \end{pmatrix} \tag{3}$$

where $n \triangleq r+s$, $d(z) \triangleq Adj(a(z))b(z)L'$, L is the Cholesky factor of Σ_ϵ, that is, an upper triangular matrix such that $L'L = \Sigma_\epsilon$ and $\hat{\epsilon}_t = (L^{-1})'\epsilon_t$ is a standardized white noise process.

Using the Wiener-Kolmogorov formula that assumes that we have a doubly infinite realization of y_t, (e.g. Caines(1988), p. 139), we can express the estimator of s_t in the form

$$\hat{s}_t = G_{sy}(B,F)G_y^{-1}(B,F)y_t, \tag{4}$$

where G_{sy} and G_y are the covariance generating functions of s_t with y_t and of y_t respectively, obtained from (3). Some key points follow:

(i) The pre-multiplication by $Adj(A(B))$ does not change the covariance generating function of the x_t process.

(ii) Because of the properties of the conditional expectations (or, equivalently, of Hilbert space projections), if we put the finite sample estimates $\hat{y}_t$ instead of y_t in (4), we get the finite sample estimates of s_t, that we will also call $\hat{s}_t$. The meaning of $\hat{y}_t$ is y_t itself for $t \in \{1, \cdots, T\}$ and the optimal finite sample forecasts or backcasts for $t > T$ or $t \leq 0$ respectively. Later we will explain how to obtain these quantities, but it should be emphasized that we only need a small number of forecasts and backcasts.

(iii) Since we have a diagonal AR part in (3), the AR terms in (4) cancel out, obtaining a filter that is simple enough to be computed in practice.

The joint and cross-covariance generating functions of s_t and y_t are, respectively

$$\begin{aligned} G(z) =& (det(a(z))^{-1}d(z)d'(z^{-1})(det(a(z^{-1}))^{-1} \text{ and} \\ G_{sy}(z) =& (det(a(z))^{-1}[d_{11}(z), d_{12}(z)] \begin{bmatrix} d'_{21}(z^{-1}) \\ d'_{22}(z^{-1}) \end{bmatrix} (det(a(z^{-1}))^{-1}, \end{aligned}$$

and the marginal generating function of y_t is

$$\begin{aligned} G_y(z) =& (det(a(z))^{-1}[d_{21}(z), d_{22}(z)] \begin{bmatrix} d'_{21}(z^{-1}) \\ d'_{22}(z^{-1}) \end{bmatrix} (det(a(z^{-1}))^{-1} \\ =& (det(a(z))^{-1}\Omega(z)\Omega'(z^{-1})(det(a(z^{-1}))^{-1}, \end{aligned}$$

that is, we have made a (left) spectral factorization of the central factors in $G_y(z)$.

Therefore, the optimal filter is

$$\hat{s}_t = [d_{11}(B), d_{12}(B)] \begin{bmatrix} d'_{21}(F) \\ d'_{22}(F) \end{bmatrix} \Omega'^{-1}(F)\Omega^{-1}(B)\hat{y}_t, \tag{5}$$

and we can compute the exact finite Wiener Kolmogorov filter running three cascaded filters, these are $\hat{f}_{1t} = \Omega^{-1}(B)\hat{y}_t$, with time running forwards, $\hat{f}_{2t} = \tilde{\Omega}'^{-1}(F)\tilde{e}_0(F)\hat{f}_{1t}$, with time running backwards, where $e'_0(z) \triangleq [d_{21}(z), d_{22}(z)]$ and $e_0(z^{-1})\Omega'^{-1}(z^{-1}) = \tilde{\Omega}'^{-1}(z^{-1})\tilde{e}_0(z^{-1})$

(we transform a right matrix fraction into a left coprime matrix fraction), and finally $\hat{s}_t = [d_{11}(B), d_{12}(B)]\hat{f}_{2t}$ with time running forwards.

The problem of finding the initial values for these or other filters is solved in section 4, but now we will make some additional matrix polynomial computations that will allow us to reduce the three filters to two new filters, one running backwards in time and the other one running forwards in time.

We can write $G_y(z) = (det(a(z))^{-1}\Omega_0(z^{-1})\Omega'_0(z)(det(a(z^{-1}))^{-1}$, that is, we apply right, instead of left spectral factorization. With this change, the filter can be expressed as $\hat{s}_t = [d_{11}(B), d_{12}(B)]\begin{bmatrix} d'_{21}(F) \\ d'_{22}(F) \end{bmatrix}\Omega_0'^{-1}(B)\Omega_0^{-1}(F)\hat{y}_t$. These are in fact four cascaded filters, so apparently the filter is more complicated than before, but defining $d_0(z) \triangleq [d_{11}(z), d_{12}(z)]$, and transforming right matrix fraction descriptions into left ones (or v.v.) we have

$$
\begin{aligned}
\hat{s}_t &= d_0(B)e_0(F)\Omega_0'^{-1}(B)\Omega_0^{-1}(F)\hat{y}_t \\
&= d_0(B)\Omega_1'^{-1}(B)e_1(F)\Omega_0^{-1}(F)B^{h_1}\hat{y}_t \qquad (6) \\
&= \Omega_2'^{-1}(B)d_1(B)e_1(F)\Omega_0^{-1}(F)B^{h_1}\hat{y}_t \qquad (7) \\
&= \Omega_2'^{-1}(B)d_1(B)\Omega_3^{-1}(F)e_2(F)B^{h_1}\hat{y}_t \qquad (8) \\
&= \Omega_2'^{-1}(B)\Omega_4^{-1}(F)d_2(B)e_2(F)B^{h_2}\hat{y}_t \qquad (9) \\
&= \Omega_2'^{-1}(B)d_3(B)\Omega_5^{-1}(F)e_2(F)B^{h_3}\hat{y}_t. \qquad (10) \\
&= \Omega_2'^{-1}(B)d_4(B)\Omega_5^{-1}(F)e_3(F)\hat{y}_t. \qquad (11)
\end{aligned}
$$

The notation that we have used increases the subscript of a polynomial matrix whenever it is involved into a transformation of a right matrix fraction into a left coprime matrix fraction (or a left one into a right coprime one). It is to be noted that we have made these transformations not only when both matrices, denominator and numerator, are written in the same operator (F or B), but also when they are written in different operators. This can be done taking into account that $F = B^{-1}$, the details are in Aparicio-Perez (2011), the only difference with the case in which both operators are equal is that now a positive or negative integer power of B can result, and that is the reason for the B^{h_j} terms that appear in (6)-(11). Finally, in (11), if $h_3 > 0$, $d_4(B) = d_3(B)B^{h_3}$ and $e_3(F) = e_2(F)$, while if $h_3 < 0$, $e_3(F) = e_2(F)F^{-h_3}$ and $d_4(B) = d_3(B)$. Something similar can be done with B^{h_2} in (9).

We will use formulae (9) and (11) in section 5, the latter being preferred to (8) to ensure coprimeness in the filters. In (11) we have two cascaded filters, the first one with time running backwards and the second one with time running forwards, the filters are

$$\Omega_5(B)\breve{v}_s = e_3(B)\breve{y}_s, \qquad (12)$$

$$\Omega'_2(B)\hat{s}_t = d_4(B)v_t, \qquad (13)$$

where v_t is a new process, while (9) is the following central difference equation that will allow us to compute the Wiener-Kolmogorov filter as a system of linear equations:

$$\Omega_4(F)\Omega'_2(B)\hat{s}_t = d_2(B)e_2(F)B^{h_2}y_t. \qquad (14)$$

Example 1. *Consider the following VARMA system of dimension four, where only two components are observed:*

$$\begin{pmatrix} 1-0.2B & 0 & 0 & 0.3B \\ 0 & 1 & 0.5B & 0 \\ 0.6B & 0 & 1-0.4B & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} s_{1t} \\ s_{2t} \\ y_{1t} \\ y_{2t} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -0.1B & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.7B \\ 0 & 0 & 0 & 1-0.8B \end{pmatrix} \begin{pmatrix} \epsilon_{1t} \\ \epsilon_{2t} \\ \epsilon_{3t} \\ \epsilon_{4t} \end{pmatrix}, \qquad (15)$$

with $cov(\epsilon) = I_4$. The resulting polynomial matrices (with two decimal places) are $d_0(B) =$

$$\begin{pmatrix} 1-.4B & 0 & 0 & -.3+.36B-.096B^2 \\ -.1B+.36B^2-.008B^3 & 1-.6B+.08B^2 & -.5B+.1B^2 & -.35B-.02B^2+.072B^3 \end{pmatrix},$$

$$e_0(F) = \begin{pmatrix} -.6F & 0 \\ 0 & 0 \\ 1-.2F & 0 \\ .7F+.04F^2-.144F^3 & 1-1.4F+.56F^2-.064F^3 \end{pmatrix},$$

$$\Omega_0(z) = \begin{pmatrix} 1.34-.12z-.075z^2 & .3z-.08z^2 \\ .33+.272z+.2z^2-.11z^3 & .87-1.35z+.68z^2-.11z^3 \end{pmatrix},$$

$$\Omega_2'(B) = \begin{pmatrix} 1-.95B & 0 \\ .17-.062B & 1 \end{pmatrix}, e_2(F) = \begin{pmatrix} .59-.44F & .31 \\ 0 & 0 \\ 0 & 0 \\ 1.92 & 1 \end{pmatrix}, d_2(B) =$$

$$\begin{pmatrix} 1+.33B-.79B^2+.77B^3 & 0 & 0 & .044-.36B+.42B^2-.23B^3 \\ .17+.057B+1.01B^2 & 1-.6B+.08B^2 & -.5B+.1B^2 & .0075-.061B-.31B^2 \end{pmatrix},$$

$$\Omega_4(F) = \begin{pmatrix} 1-.15F & 0 \\ -.025F & 1 \end{pmatrix}, e_4(F) = \begin{pmatrix} .23+.39F & .21 \\ 1.28+.82F & -1.18 \\ 2.56F+1.63F^2 & -2.36F \\ 1.92+1.99F+1.32F^2 & -1.77-.71F \end{pmatrix},$$

$$d_3(B) = \begin{pmatrix} 1.31-1.27B+.77B^2 & 0 & 0 & .09B-.092B^2 \\ -.57+1.03B & -.4+.08B & 0 & -.18B \end{pmatrix},$$

$$\Omega_5(F) = \begin{pmatrix} 1-.77F+.091F^2 & 0 & 0 & .19-.027F \\ 2.22F & 1+2.5F & -1.25 & -.67 \\ 4.45F^2 & 5F^2 & 1-2.5F & -1.33F \\ -2.07F^2 & 0 & 0 & 1+.62F \end{pmatrix},$$

$h_1 = 0$, $h_2 = 0$, $h_3 = 1$, $e_3(F) = e_2(F)$ and $d_4(B) = Bd_3(B)$. ◇

Parallel implementations of the filter are also possible. For example, we can transform (11) into

$$\hat{s}_t = \Omega_2'^{-1}(B)d_4(B)e_4(F)\Omega_6^{-1}(F)\hat{y}_t, \tag{16}$$

and this filter can be expressed as the sum of two filters, the first one is causal, it is applied to the original y_t process with time running forwards, while the second one is anti-causal, it is applied to the time reversed $\breve{y}_s$ process with time running backwards. To do so, we have to solve the equation $\Omega_2'^{-1}(z)d_4(z)e_4(z^{-1})\Omega_6^{-1}(z^{-1}) = \Omega_2'^{-1}(z)g(z) + h(z^{-1})\Omega_6^{-1}(z^{-1})$ for $g(z)$ and $h(z^{-1})$. Pre-multiplying this expression by $\Omega_2'(z)$ and post-multiplying the result by $\Omega_6(z^{-1})$ we obtain the bilateral polynomial equation

$$d_5(z)e_1(z^{-1}) = g(z)\Omega_0(z^{-1}) + \Omega_6'(z)h(z^{-1}) \tag{17}$$

and the $g(z)$ and $h(z^{-1})$ polynomial matrices can be obtained by equating the coefficients of the corresponding powers of z and z^{-1}. The question arises as if (17) always has a solution, and the answer is in the affirmative, because the polynomial matrices $\Omega_6(z^{-1})$ and $\Omega_2'(z)$ have no common factors, since $\Omega_6(z^{-1})$ has the same Smith form as $\Omega_0(z^{-1})$, see Lemma 1 in Roberts and Newmann (1988).

The procedure that we have used to solve (17) is similar to the one explained in the appendix in Burman(1980) for the univariate case.

The filtering problem is then solved by running the two filters and summing, that is

$$\Omega_2'(B)s_t^1 = g(B)y_t, \tag{18}$$

$$\Omega_7(B)\breve{s}_s^2 = h_2(B)\breve{y}_s, \tag{19}$$

$$\hat{s}_t = \hat{s}_t^1 + \hat{s}_t^2, \tag{20}$$

where $h(z)\Omega_6^{-1}(z) = \Omega_7^{-1}(z)h_2(z)$.

Moreover, the parallel implementation contains some additional information, since it decomposes the transfer function of the Wiener-Kolmogorov filter as the sum of its causal and non-causal components, allowing us to compute the filter weights by long division in (18)-(19).

We could also define a parallel implementation directly from (7), instead of (11), but lack of left-coprimeness between the factors in (7) would make that the equations, similar to (17), to be solved had a larger number of unknowns and, finally, the two resulting parallel filters would have to be transformed into left-coprime ones. The lack of coprimeness between the factors in (7) comes from the pre-multiplication of (1) by $Adj(a(B))$ to obtain (3)

Example 1 (Continued). *The polynomial matrices that define the parallel filters are:*

$$\Omega_6(F) = \begin{pmatrix} 1+1.66F & -1.44 \\ 2.07F^2 & 1-1.81F \end{pmatrix}, h(F) = \begin{pmatrix} .25 & 0 \\ -.0036 & 0 \end{pmatrix},$$

$$\Omega_7(F) = \begin{pmatrix} 1-.15F & 0 \\ .014 & 1 \end{pmatrix}, h_2(F) = \begin{pmatrix} .25-.46F & .37 \\ 0 & 0 \end{pmatrix},$$

$$\textit{and } g(B) = \begin{pmatrix} -.14 + .15B - .030B^2 & .11B + .13B^2 \\ .02 - .51B & -.016B \end{pmatrix}.$$

◇

We have solved the non-causal filtering problem, but the causal filtering problem can be solved in a similar way, provided that we compute at each time point the exact corresponding $\hat{y}_t$ values.

3 An extended innovations algorithm

The innovations algorithm is a well known technique that can be used to efficiently compute the exact finite sample innovations and one-step predictions of a multivariate time series, see e.g. Brockwell and Davis (1991), secs. 5.3 and 11.4. In this section we will present an extension of this algorithm that allows the estimation of any other random variables, conditional on the observed time series, provided that we know its cross-covariances with the time series. In section 4.1 we will use this extension to compute the initial and final values of the filters that realize the Wiener-Kolmogorov filter.

First we will present the idea behind the extended innovations algorithm as a problem of computing the conditional distribution of a multivariate normal random vector without having to compute the inverse of a big covariance matrix.

Let ψ be a vector of random variables, that we consider partitioned in the form $\psi' = (\psi_1', \psi_2')'$. Assume that ψ follows the normal distribution $N\left(\left(\begin{smallmatrix}\mu_1\\ \mu_2\end{smallmatrix}\right), \left(\begin{smallmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{21} & \Sigma_{22}\end{smallmatrix}\right)\right)$ where the mean vector and covariance matrix are partitioned in accordance with the partition of ψ. It is well known that the conditional distribution of ψ_1, given the value of ψ_2, is normal with conditional mean and covariance matrix given by:

$$\psi_1/\psi_2 \sim N(\mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(\psi_2 - \mu_2), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}), \tag{21}$$

that the conditional mean is the best predictor of ψ_1 as a function of ψ_2 in the least square sense, and that it has a mean square error (MSE) given by the conditional covariance matrix. This principle can be applied to filtering and smoothing of multivariate time series. For example, given the model (1) we can compute the autocovariances of the x_t process, and fill with them the (block Toeplitz) covariance matrix of $x_{1:T}$, where $x_{1:T}' \triangleq (x_1', \cdots, x_T')'$, this is

$$cov(x_{1:T}) = \begin{pmatrix} \Delta_0 & \Delta_1' & \cdots & \Delta_{T-1}' \\ \Delta_1 & \Delta_0 & \cdots & \Delta_{T-2}' \\ \cdots & \cdots & \cdots & \cdots \\ \Delta_{T-1} & \Delta_{T-2} & \cdots & \Delta_0 \end{pmatrix}, \tag{22}$$

with $\Delta_h \triangleq cov(x_{t+h}, x_t)$, $h \geq 0$. Defining $\psi_1' \triangleq s_{1:T}' \triangleq (s_1', \cdots, s_T')'$ and $\psi_2' \triangleq y_{1:T}' \triangleq (y_1', \cdots, y_T')'$, the mean vector of ψ is zero and its covariance matrix Σ is built by permuting some rows and columns in (22).

With this setting, the exact Wiener-Kolmogorov filter that estimates ψ_1, given ψ_2 is simply obtained using (21). Assuming that the model (1) is fixed, the dimension of Σ_{22} grows with the number of observations T, and its inversion soon becomes a complex numerical problem, with a number of operations that is $O(T^3)$. Even taking into account the block Toeplitz structure of (22) the number of operations is $O(T^2)$.

A solution is to transform the ψ_2 vector into another vector, say ψ_2^*, such that the covariance matrix of ψ_2^* is a block diagonal matrix, because the inversion of a block diagonal matrix is very simple: it is enough to invert each block in the diagonal. A transformation of this kind is equivalent to a Gram-Schmidt ortogonalization, since the vectors of random variables with finite second order moments can be considered to form a vector space. And this is what the innovations algorithm does in its basic version, to orthogonalize the random vectors $y_1, \cdots, y_T$. Unfortunately, the complexity of the Gram-Schmidt procedure grows quickly with the number of vectors that are orthogonalized, since the projection of each new vector over all the preceding ones has to be computed. For this reason, the innovations algorithm, when applied to a VARMA process, makes a previous step: it first transforms the original process y_t into a new process w_t that is only finitely correlated, that is, in geometric terms, each vector to be orthogonalized is already orthogonal to all the preceding vectors except the last q ones, where q is the order of the MA part of the marginal model for y_t. The output of the Gram-Schmidt procedure is the innovations process $I_t \triangleq w_t - \hat{w}_{t/t-1}$. From these quantities, the $\hat{y}_t$ and the innovations of the y_t process are easily computed.

However, the innovations algorithm considers only the ψ_2 vector formed by the w_t process, and it only computes the innovations of the w_t and y_t processes. The proposed extension considers a ψ_1 vector and estimates it using its cross-covariances Σ_{12} with the y_t process. In our time series setting ψ_1 would be formed by the $s_1, \cdots, s_T$ values to be estimated. Still, this would create a new problem because, as T grows, we would have to compute a cross-covariance matrix Σ_{12} of growing dimension and update it in each iteration of the innovations algorithm and the number of operatios would be $O(T^2)$ again. The solution to this problem is to define another ψ_1 vector that has a small and fixed (not growing with T) dimension, but that is enough to compute the Wiener-Kolmogorov filter. For example, if we were using the parallel filters (18)-(19), we could define ψ_1 to contain only the initial values of y_t and s_t^1 and the final values of y_t and s_t^2. With these initial and final values, the two parallel filters could easily be run to estimate $s_1, \cdots, s_T$. The number of operations of this procedure grows only linearly with T, since as the sample size increases from T to $T+1$ we only have to update the covariances of a fixed number of random variables with a new innovation and then run the innovations algorithm and the two filters at another time point.

We will now give the details of the extended innovations algorithm. Assume that the observed process y_t follows the model (2), with $\Phi_o = I_r$ (if this is not the case we previously pre-multiply (2) by Φ_o^{-1}). Assume also that we have ψ_1, a vector of random variables whose covariance with y_t, that we call C_t, is known for all $t \in \{1, \cdots, T\}$ and that the joint distribution of ψ_1 and $y_{1:T}$ is multivariate normal, as it will be the case, for example, if ψ_1 is composed of some elements of the s_t and y_t processes that follow the gaussian model (1).

Initially, we have $MSE(\hat{\psi}_1) = cov(\psi_1)$, because the best prediction of ψ_1 when we do not have any information is its mean. We will update $MSE(\hat{\psi}_1)$ as we iterate in the algorithm. We define $\Sigma_{12} \triangleq [C_1, \cdots, C_T]$. Let $p = \partial(\Phi(z))$ and $q = \partial(\Theta(z)$ be the degrees of the AR and MA polynomial matrices respectively, $r \triangleq max(p,q)$ and $\Gamma_h \triangleq cov(y_{t+h}, y_t)$, for h integer. We now define w_t, the y_t process filtered by its own autoregressive part, that is $w_t = y_t$, for $t \in \{1, \cdots, r\}$ and $w_t = \Phi(B)y_t$, for $t \in \{r+1, \cdots, T\}$. This process is only finitely correlated, in fact, if $i \geq j$ are integers, $h \triangleq i-j$ and $\gamma_{i,j} \triangleq cov(w_i w_j)$,

$$\gamma_{i,j} = \begin{cases} 0 & \text{if } h > q, \\ \Gamma_h & \text{if } i \leq r, \\ \sum_{k=0}^{p} \Phi_k \Gamma_{h-k} & \text{if } i > r, j \leq r \text{ and } h \leq q \\ \sum_{k=h}^{q} B_k \Sigma_u B'_{k-h} & \text{if } j > r \text{ and } h \leq q \end{cases} \text{, and for } i < j, \gamma_{i,j} = \gamma'_{j,i}.$$

If $w'_{1:T} \triangleq (w'_1, \cdots, w'_T)'$, we first compute $\Sigma^*_{12} \triangleq cov(\psi_1, w_{1:T}) \triangleq [C^*_1, \cdots, C^*_T]$, using $C^*_t = \sum_{j=1}^{p} C_{t-j}\Phi_j$, for $t \in \{r+1, \cdots, T\}$, and $C^*_t = C_t$ for $t \in \{1, \cdots, r\}$.

Defining $V_j \triangleq cov(I_j)$ we have, initially, $\hat{w}_1 \triangleq 0$, $\hat{I}_1 \triangleq w_1$ and $V_1 = cov(w_1)$.

The main iteration begins with i ranging from 1 to T, for i fixed we do:

1. Compute $T_{ij} \triangleq cov(w_i, I_j) = \gamma_{i,j} - \sum_{k=max\{1,i-q\}}^{i-1} T_{i,k} V_k^{-1} T'_{j,k}$ and $T^*_{ij} \triangleq cov(w_i, I_j)V_j^{-1}$, for each $j \in \{max\{i-q, 1\}, \cdots, i-1\}$,

2. Compute $\hat{w}_i = \sum_{j=max\{i-q,1\}}^{i-1} T^*_{ij} I_j$, and $\hat{y}_i = \begin{cases} \hat{w}_i - \sum_{j=1}^{p} \Phi_j y_{i-j} & \text{if } i > r \\ \hat{w}_i & \text{if } i \leq r \end{cases}$, the one-step ahead forecasts of w_i and y_i, respectively.

3. Compute the innovation $I_i = w_i - \hat{w}_i$ and $V_i = \gamma_{i,i} - \sum_{k=max\{1,i-q\}}^{i-1} T_{i,k} V_k^{-1} T'_{i,k}$.

4. Update $cov(\psi_1, I_i) = C^*_i - \sum_{k=max(1,i-q)}^{i-1} cov(\psi_1, I_k) T^{*\prime}_{ik}$, $\hat{\psi}_1 = \hat{\psi}_1 + cov(\psi_1, I_i)V_i^{-1} I_i$ and $MSE(\hat{\psi}_1) = MSE(\hat{\psi}_1) - cov(\psi_1, I_i)V_i^{-1}(cov(\psi_1, I_i))'$.

As we said before, the computational effort added to the innovations algorithm is small for a fixed dimension of ψ_1. In iteration i only the computation of $cov(\psi_1, I_i)$ and the updates are added, and, since $\hat{w}_i$ is function of only the last q innovations, this involves few operations. As a result, the number of operations involved in the extended innovations algorithm is still $O(T)$.

Example 2. *A VARMA system of dimension two is given:*

$$\begin{pmatrix} 1-.7B & 0 \\ 0 & 1-.6B \end{pmatrix} \begin{pmatrix} s_t \\ y_t \end{pmatrix} = \begin{pmatrix} 1+.5B & .6B \\ -.7B & 1+.8B \end{pmatrix} \begin{pmatrix} \epsilon_{1t} \\ \epsilon_{2t} \end{pmatrix}, \Sigma_\epsilon = \begin{pmatrix} 1 & .71 \\ .71 & 2 \end{pmatrix} \tag{23}$$

We have a sample of size 3 of y_t, $y_{1:3} = (1, -1, .5)'$ and we will use the extended innovations algorithm to estimate $\psi_1 = (s_0, y_0, s_4)'$. Using two decimal places, the marginal model of y_t is

$(1-.6B)y_t = (1+.44B)u_t$, with $\Sigma_u = 2.49$, the autocovariances of the joint process (23) and the covariances of ψ_1 with $y_{1:3}$ and $w_{1:3}$ are

$$\Delta_0 = \begin{pmatrix} 7.24 & 3.7 \\ 3.7 & 6.72 \end{pmatrix}, \Delta_1 = \begin{pmatrix} 5.99 & 4.15 \\ 2.09 & 5.13 \end{pmatrix}, \Delta_2 = \begin{pmatrix} 4.2 & 2.9 \\ 1.25 & 3.08 \end{pmatrix}, \Delta_3 = \begin{pmatrix} 2.94 & 2.03 \\ .75 & 1.85 \end{pmatrix},$$

$$\Delta_4 = \begin{pmatrix} 2.06 & 1.42 \\ .45 & 1.11 \end{pmatrix}, \Sigma_{12} = \begin{pmatrix} 2.09 & 1.25 & .75 \\ 5.13 & 3.08 & 1.85 \\ 2.03 & 2.9 & 4.15 \end{pmatrix}, \Sigma_{12}^* = \begin{pmatrix} 2.09 & 0 & 0 \\ 5.13 & 0 & 0 \\ 2.03 & 1.68 & 2.4 \end{pmatrix},$$

we also have $\gamma_{i,j} \triangleq cov(w_i, w_j) = \begin{cases} 0 & \text{if } h > 1, \\ 6.72 & \text{if } i = j = 1, \\ 1.1 & \text{if } i = j+1, \\ 2.97 & \text{if } i = j > 1. \end{cases}$

We define $\hat{\psi}_{1/0} = (0,0,0)'$, $MSE(\hat{\psi}_{1/0}) = \begin{pmatrix} 7.24 & 3.7 & 2.06 \\ 3.7 & 6.72 & 1.42 \\ 2.06 & 1.42 & 7.24 \end{pmatrix}$, *and the main iteration begins:*

- *For $i = 1$, $\hat{w}_1 = 0$, $I_1 = 1$, $V_1 = 6.72$, $\hat{\psi}_{1/1} = \hat{\psi}_{1/0} + C_1^* V_1^{-1} I_1 = (.31, .76, .3)'$.*

  $$MSE(\hat{\psi}_{1/1}) = MSE(\hat{\psi}_{1/0}) - C_1^* V_1^{-1} C_1^{*\prime} = \begin{pmatrix} 6.59 & 2.1 & 1.42 \\ 2.1 & 2.79 & -.13 \\ 1.42 & -.13 & 6.63 \end{pmatrix}.$$

- *For $i = 2$ and $j = 1$, $T_{21} = \gamma_{2,1} = 1.1$, $T_{21}^* = .16$, $\hat{w}_2 = .16$, $w_2 = -1.6$, $\hat{y}_2 = .76$, $I_2 = -1.76$, $V_2 = 2.79$, $cov(\psi_1, I_2) = C_2^* - C_1^* T_{21}^{*\prime} = (-.34, -.84, 1.35)'$, $\hat{\psi}_{1/2} = \hat{\psi}_{1/1} + cov(\psi_1, I_2) V_2^{-1} I_2 = (.53, 1.3, -.55)'$.*

  $$MSE(\hat{\psi}_{1/2}) = MSE(\hat{\psi}_{1/1}) - cov(\psi_1, I_2) V_2^{-1} cov(\psi_1, I_2)^{*\prime} = \begin{pmatrix} 6.55 & 2 & 1.59 \\ 2 & 2.54 & .28 \\ 1.59 & .28 & 5.97 \end{pmatrix}.$$

- *Finally $i = 3$ and $j = 2$. $T_{32} = \gamma_{3,2} = 1.1$, $T_{32}^* = .4$, $\hat{w}_3 = -.7$, $w_3 = 1.1$, $\hat{y}_3 = -1.3$, $I_3 = 1.8$, $V_3 = 2.54$, $cov(\psi_1, I_3) = C_3^* - cov(\psi_1, I_2) T_{32}^{*\prime} = (.14, .33, 1.87)'$, $\hat{\psi}_{1/3} = \hat{\psi}_{1/2} + cov(\psi_1, I_3) V_3^{-1} I_3 = (.62, 1.53, .77)'$.*

  $$MSE(\hat{\psi}_{1/3}) = MSE(\hat{\psi}_{1/3}) - cov(\psi_1, I_3) V_3^{-1} cov(\psi_1, I_3)^{*\prime} = \begin{pmatrix} 6.54 & 1.98 & 1.49 \\ 1.98 & 2.5 & .031 \\ 1.49 & .031 & 4.59 \end{pmatrix}.$$

◇

4 Computation of the initial values

In this section, two techniques are developed that can be used to compute the initial values of a general rational filter. Both techniques are new.

We will assume that we have a process y_t, of dimension r that follows a regular, gaussian, left-coprime, stationary and invertible model of the form (2), and another m-dimensional process z_t that is defined as the following rational filter applied to y_t:

$$D(B)z_t = N(B)y_t. \tag{24}$$

$D(z)$ is assumed to be a non-singular and stable polynomial matrix. The time reversed process $\breve{y}_s$ follows a model that can be obtained as explained in Aparicio-Perez (2011), let this model be

$$\breve{\Phi}(B)\breve{y}_s = \breve{\Theta}(B)\breve{u}_s, \tag{25}$$

with $\breve{u}_s$ a new white noise process with covariance matrix $\Sigma_{\breve{u}}$. If $g_D \triangleq \partial D(z)$ and $g_N \triangleq \partial N(z)$, then we need g_N initial values of y_t and g_D initial values of z_t to begin the iterations when computing the values of z_t in the filter (24). We will use the subscripts $_I$ for the initial and $_F$ for the final values of a process, the meaning of final values being initial values of the time-reversed process. Then $y_I' \triangleq (y'_{-g_N+1}, \cdots, y'_0)'$, $z_I' \triangleq (z'_{-g_D+1}, \cdots, z'_0)'$ and y_I can be computed simply by forecasting with its time-reversed model, but we need some procedure to compute z_I.

4.1 Using the extended innovations algorithm

The extended innovations algorithm can be used to compute, using the observed data $y_{1:T}$ and in only one run, $\hat{y}_I$ and $\hat{z}_I$. In the notation of section 3 we have $\psi_1' = (z_I', y_I')' = (z'_{-g_D+1}, \cdots, z'_0, y'_{-g_N+1}, \cdots, y'_0)'$ and we only need to compute the covariances of y_I and z_I with $y_{1:T}$. To do so we first need to obtain the model that follows the joint process formed by z_t and y_t, this is the model that follows the filtered y_t process, using the rational filter

$$\begin{pmatrix} D(B) & 0 \\ 0 & I_r \end{pmatrix} \begin{pmatrix} z_t \\ y_t \end{pmatrix} = \begin{pmatrix} N(B) \\ I_r \end{pmatrix} y_t, \tag{26}$$

and then we have to compute the autocovariances of this joint process. Of course the joint process cannot be regular, since all randomness comes from the white noise of the y_t process, but this fact does not pose any additional problems if the methods explained in Aparicio-Perez (2011) to find the joint VARMA process and its autocovariances are used.

Example 1 (Continued). *Assume that we have three observations of y_t, these are $y_1 = (1, 0.9)'$, $y_2 = (-0.5, -0.4)'$ and $y_3 = (0.8, 0.7)'$. The parallel backwards filter is*

$$\begin{pmatrix} 1 - .15B & 0 \\ .014 & 1 \end{pmatrix} \breve{s}_s^2 = \begin{pmatrix} .25 - .46B & .37 \\ 0 & 0 \end{pmatrix} \breve{y}_s.$$

Since the degrees of the filter numerator and denominator are both 1, we have $s_F^2 = s_4^2$, $y_F = y_4$ and $\psi_1' = (s_4^{2\prime}, y_4')'$. The backwards marginal model for $\breve{y}_s$ is (25) with

$$\breve{\Phi}(B) = \begin{pmatrix} 1 - .6B + .05B^2 & 0 \\ 6.87 - 2.68B & 1 \end{pmatrix}, \breve{\Theta}(B) = \begin{pmatrix} .25 - .46B & .37 \\ 0 & 0 \end{pmatrix}, \Sigma_{\breve{u}} = \begin{pmatrix} 1.8 & -.44 \\ -.44 & .87 \end{pmatrix}$$

and the joint (singular) process of $\breve{s}_s^2$ and $\breve{y}_s$ can be written in the form

$$\begin{pmatrix} 1-.49B+.05B^2 & 0 & .052B & 0 \\ .014 & 1 & 0 & 0 \\ .15B-.022B^2 & 0 & 1-.25B & 0 \\ -5.11+.75B & 0 & -4.71 & 1 \end{pmatrix} \begin{pmatrix} \breve{s}_s^2 \\ \breve{y}_s \end{pmatrix} =$$

$$\begin{pmatrix} .25-.41B+.022B^2 & .37-.38B+.026B^2 \\ 0 & 0 \\ 1+.38B-.0098B^2 & .4B-.011B^2 \\ -6.01-1.07B & -.88-1.25B \end{pmatrix} \breve{u}_s.$$

Computing the autocovariances of this joint process, we can build

$$\Sigma_{12} = cov(\begin{pmatrix} s_4^2 \\ y_4 \end{pmatrix}, \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}) = \begin{pmatrix} .012 & .014 & .035 & .072 & .12 & -.38 \\ -.00016 & -.00019 & -.00049 & -.0010 & -.0016 & .0054 \\ .14 & .069 & .38 & .4 & 1.13 & .33 \\ 0 & 0 & 0 & 0 & 0 & -.8 \end{pmatrix}$$

and the extended innovations algorithm gives $\hat{s}_F^2 = (-.19, .0027)'$ and $\hat{y}_F = (.79, -.55)'$. $\diamond$

The application of this method to our Wiener-Kolmogorov filtering problem is straightforward:

If we are filtering using the central difference equation (14), we need y_I, y_F, s_I and s_F, and the necessary autocovariances and cross-covariances are readily obtained from the joint model (1).

If we are filtering with the two cascaded filters (12)-(13) we need y_F, v_F, v_I and s_I. The cross-covariances between v_t and y_t can be obtained from the joint backwards marginal model of $\breve{v}_s$ and $\breve{y}_s$, that we can compute from (26) and (25). The cross-covariances between s_t and y_t are computed from (1).

If we are computing the parallel filters we need y_I, y_F, s_I^1 and s_F^2 and the cross-covariances of s_I^1 and s_F^2 with y_t can be computed from (18)-(19), using (26) and both the forwards and backwards marginal models of y_t.

In any of the three cases we put in Σ_{12}, as defined in section 3, the corresponding autocovariances and cross-covariances.

4.2 A left-right matrix geometric series

To compute the initial values z_I we can sum something that could be called a left-right matrix geometric series. The next simple example motivates the development:

Example 3. *Assume scalar y_t and z_t processes, with $D(z) = 1 - az$, $N(z) = 1 - bz$, $\Phi(z) = 1 - \phi z$ and $\Theta(z) = 1 + \theta z$, where $|a| < 1$, $|b| < 1$, $|\phi| < 1$ and $|\theta| < 1$. Once we have computed $\hat{y}_0$ we can compute $\hat{y}_{-j} = \phi^j \hat{y}_0$, for all $j \geq 1$. On the other side, the filter weights in $\Psi_j(z) = D^{-1}(z)N(z)$*

are $\psi_0 = 1$, $\psi_1 = a - b$ and $\psi_j = a^{j-1}\psi_1$, for all $j \geq 2$. Then we have $\hat{z}_0 = \sum_{j=0}^{\infty} \psi_j \hat{y}_{-j} = \hat{y}_0 + \sum_{j=0}^{\infty} a^j \psi_1 \phi^j \hat{y}_{-1}$, and the infinite sum can be readily computed as a geometric series, the result is $\hat{z}_0 = \hat{y}_0 + (a-b)\phi\hat{y}_0/(1-a\phi)$. It is to be noted that the geometric series converges even if the modulus of one of a and ϕ is equal to one, provided that the modulus of the other is strictly smaller than one. ◇

The method that follows treats the general VARMA case in the same way, summing a new kind of matrix geometric series.

First, assume without loss in generality that $D_0 = I_m$ and that $\breve{\Phi}_0 = I_r$, if this is not the case, $D(z)$ and $N(z)$ are pre-multiplied by D_0^{-1}, while $\breve{\Phi}(z)$ and $\breve{\Theta}(z)$ are pre-multiplied by $\breve{\Phi}_o^{-1}$ without affecting the filter (24) or the model (25). Note that D_0 has an inverse because we assumed that the polynomial matrix $D(z)$ is not singular.

Under our assumptions, if $k \geq 0$, $\hat{z}_{-k} = \sum_{j=0}^{\infty} \Psi_j \hat{y}_{-k-j}$, where $\Psi(z) = D^{-1}(z)N(z)$ contains the filter weights.

On one side, for $k > k_0 \triangleq max(g_D, g_N)$, the filter weights follow the relation $D(B)\Psi_k = 0$, that is, we can write for $j \geq 1$, $\Psi_{k_0+j} = -\sum_{i=1}^{g_D} D_j \Psi_{k_0+j-i} = MA^j N_{k_0}$, with $M \triangleq (0, \cdots, 0, I_m)$, $A \triangleq \begin{pmatrix} 0 & I_m & 0 & \cdots & 0 \\ 0 & 0 & I_m & \cdots & 0 \\ \cdots & & & & \\ 0 & 0 & 0 & \cdots & I_m \\ -D_{g_D} & -D_{g_D-1} & -D_{g_D-2} & \cdots & -D_1 \end{pmatrix}$ and $N_i \triangleq \begin{pmatrix} \Psi_{i-g_D+1} \\ \Psi_{i-g_d+2} \\ \cdots \\ \Psi_{i-1} \\ \Psi_i. \end{pmatrix}$

On the other side, if $g_f \triangleq \partial(\breve{\Phi}(x))$, for $k > k_1 \triangleq max(\partial(\breve{\Psi}(z)), \partial(\breve{\Theta}(z)))$, the backcasted values of y_k satisfy the relation $\breve{\Phi}(B)\hat{\breve{y}}_{-k} = 0$, so we can write for $j \geq 1$, $\hat{\breve{y}}_{-k_1-j} = -\sum_{i=1}^{g_f} \breve{\Phi}_j \hat{\breve{y}}_{-k_1-j+i} = RB^j H_{-k_1-j}$, with $B \triangleq \begin{pmatrix} 0 & I_r & 0 & \cdots & 0 \\ 0 & 0 & I_r & \cdots & 0 \\ \cdots & & & & \\ 0 & 0 & 0 & \cdots & I_r \\ -\breve{\Phi}_{g_f} & -\breve{\Phi}_{g_f-1} & -\breve{\Phi}_{g_f-2} & \cdots & -\breve{\Phi}_1 \end{pmatrix}$, $R \triangleq (0, \cdots, 0, I_r)$, and $H_i \triangleq \begin{pmatrix} \hat{y}_{i+g_f-1} \\ \hat{y}_{i+g_f-2} \\ \cdots \\ \hat{y}_{i+1} \\ \hat{y}_i. \end{pmatrix}$. Therefore, if $k_2 \triangleq max(k_0, k_1)$, we can write

$$\hat{z}_{-k} = \sum_{i=0}^{k_2} \Psi_i \hat{y}_{-k-i} + \sum_{i=k_2+1}^{\infty} \Psi_i \hat{y}_{-k-i} = \sum_{i=0}^{k_2} \Psi_i \hat{y}_{-k-i} + MSH_{-k-k_2-1}, \tag{27}$$

where $S \triangleq \sum_{j=0}^{\infty} A^j N_{k_2} R B^j$.

Now, we have to compute S, but since $ASB = \sum_{j=1}^{\infty} A^j N_{k_2} R B^j$, then $N_{k_2}R = S - ASB$, applying the vec operator, $vec(S) = (I - B' \otimes A)^{-1} vec(N_{k_2}R)$.

Note that S is the same for all k, so, when computing $\hat{z}_{-k}$, for $k \in \{0, \cdots, g_D - 1\}$ only the finite sum and the H_{-k-k_2-1} vectors in (27) are different for each value of k.

The convergence of the left-right matrix geometric series that defines S can be proved taking into account the stability of the filter (24) and the model (25). In fact, it converges under the milder assumption that only one of (24) or (25) has a stable denominator, provided that the other has all its roots on or outside the unit disk. This is in accordance with the remark that we made in example 3 and it will be used in section 7.

Example 1 (Continued)**.** *We will compute the initial values of the parallel backwards filter* $\Omega_7(B)\breve{s}_{2s} = h_2(B)\breve{y}_s$ *with*

$$\Omega_7(z) = \begin{pmatrix} 1 - .15z & 0 \\ .014 & 1 \end{pmatrix} \text{ and } h_2(z) = \begin{pmatrix} .25 - .46z & .37 \\ 0 & 0 \end{pmatrix}.$$

Since this is a backwards filter, the backwards marginal model of $\breve{y}_s$ *is really the forwards marginal model of* y_t*, given by* $\Phi(B)y_t = \Theta(B)u_t$ *with*

$$\Phi(z) = \begin{pmatrix} 1 - .6z + .08z^2 & 0 \\ 0 & 1 \end{pmatrix}, \Theta(z) = \begin{pmatrix} 1 - .15z & .7z + .04z^2 - .144z^3 \\ 0 & 1 - .8z \end{pmatrix} \text{ and } \Sigma_u = \begin{pmatrix} 1.37 & 0 \\ 0 & 1 \end{pmatrix}.$$

We have $k_0 = 1$*,* $k_2 = k_1 = 3$*,* $\hat{y}_4 = \hat{\breve{y}}_0 = (0.79391, -0.54741)'$*,* $\hat{y}_5 = \hat{\breve{y}}_{-1} = (.44255, 0)'$*,* $\hat{y}_6 = \hat{\breve{y}}_{-2} = (.10348, 0)'$*,* $\hat{y}_7 = \hat{\breve{y}}_{-3} = (.026686, 0)'$*.*

Since $\partial(D(z)) = 1$*, we need only one initial value* $\hat{z}_0 = \sum_{i=0}^{3} \Psi_i \hat{y}_{-i} + MSH_{-4}$*, where* $M = I_2$*,* $H_{-4} = (\hat{y}_6', \hat{y}_7',)'$*,* $S = \sum_{j=0}^{\infty} A^j N_3 R B^j$*,* $A = \begin{pmatrix} .15 & 0 \\ -.0021 & 0 \end{pmatrix}$*,* $N_3 = \begin{pmatrix} -.0090 & .0011 \\ .00013 & -1.61 \cdot 10^{-5} \end{pmatrix}$*,* $R = (0_{2\times 2}, I_2)$*,* $B = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -.08 & 0 & .6 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$*. First we compute the finite sum* $\sum_{i=0}^{3} \Psi_i \hat{y}_{-i} = (-.19187, .0027176)'$*, then we compute* $S = \begin{pmatrix} .00011445 & 0 & -.0098056 & .0011383 \\ -1.6211 \cdot 10^{-6} & 0 & .00013889 & -1.6122 \cdot 10^{-5} \end{pmatrix}$ *and finally* $\hat{s}_F^2 = \hat{\breve{z}}_0 = (-.19212, .0027211)'$ *as before.* ◇

5 Running the filters

5.1 Simultaneous implementation: a linear system of equations

The Wiener-Kolmogorov filter can be computed using the central difference equation (14). This reduces to a linear system of equations if we previously compute y_I, y_F, s_I and s_F, for example, using the extended innovations algorithm, and then set all the values $t \in \{1, \cdots, T\}$ in (14). The unknowns are then $\hat{s}_1, \cdots, \hat{s}_T$

The number of equations and unknowns in the system grows with the length of the time series and with the dimension of s_t, so it would seem that this method can only be used for small values of T and s. However, the system of equations is block banded and the denominators $\Omega_2(z)$ and $\Omega_4(z)$ are stable polynomial matrices, since they have the same

Smith form as $\Omega_0(z)$. This means that quite large systems can be solved using fast iterative techniques.

Example 1 (Continued)**.** *In this example the central difference equation (14) is* $P_{-2}s_{t-2}+P_{-1}s_{t-1}+P_0s_t+P_1s_{t+1}=Q_{-2}y_{t-2}+Q_{-1}y_{t-1}+Q_0y_t+Q_1y_{t+1}$*, with*

$$P_{-2}=\begin{pmatrix} .12 & 0 \\ 0 & 0 \end{pmatrix}, P_{-1}=\begin{pmatrix} -.96 & 0 \\ -.065 & 0 \end{pmatrix}, P_0=\begin{pmatrix} 1.14 & 0 \\ 0.2 & 1 \end{pmatrix}, P_1=\begin{pmatrix} -.15 & 0 \\ -.025 & 0 \end{pmatrix},$$

$$Q_{-2}=\begin{pmatrix} 0 & .18 \\ 0 & 0 \end{pmatrix}, Q_{-1}=\begin{pmatrix} -.14 & -.25 \\ -.52 & -.04 \end{pmatrix}, Q_0=\begin{pmatrix} .53 & .35 \\ .09 & .06 \end{pmatrix}, Q_1=\begin{pmatrix} -.44 & 0 \\ -.075 & 0 \end{pmatrix},$$

and the linear system of equations is

$$\begin{pmatrix} P_0 & P_1 & 0 \\ P_{-1} & P_0 & P_1 \\ P_{-2} & P_{-1} & P_0 \end{pmatrix}\begin{pmatrix} \hat{s}_1 \\ \hat{s}_2 \\ \hat{s}_3 \end{pmatrix}=\begin{pmatrix} Q_{-2}\hat{y}_{-1}+Q_{-1}\hat{y}_0+Q_0y_1+Q_1y_2-P_{-2}\hat{s}_{-1}-P_{-1}\hat{s}_0 \\ Q_{-2}\hat{y}_0+Q_{-1}y_1+Q_0y_2+Q_1y_3-P_{-2}\hat{s}_0 \\ Q_{-2}\hat{y}_1+Q_{-1}y_2+Q_0y_3+Q_1\hat{y}_4-P_1\hat{s}_4 \end{pmatrix}.$$

The extended innovations algorithm provides the values $\hat{s}_{-1}=(-.29,-.039)'$, $\hat{s}_0=(-.51,-.12)'$, $\hat{s}_4=(-.21,-.4)'$, $\hat{y}_{-1}=(.28,.35)'$, $\hat{y}_0=(.86,-.86)'$, $\hat{y}_4=(.79,-.55)'$ *and the right hand side of the linear system of equations becomes* $(.76,-.27,-1.21,-.69,.62,.33)'$*. The solution is* $\hat{s}_1=(.60,-.40)'$, $\hat{s}_2=(-.56,-.54)'$ *and* $\hat{s}_3=(.0085,.29)'$. ◇

5.2 Cascaded implementation

Several possibilities arise. In all of them we need to have $\hat{y}_F$. Then, we have to compute $\hat{v}_F$ to run the first filter, any of the methods in section 4 can be used. We can run the first filter from $s=1$ to T provided that we already have $\hat{v}_I$, that we need for the second filter. If we do not have $\hat{v}_I$, then, the first filter must be run from $s=1$ to $T+\partial(d_4(z))$ and we also need $\hat{y}_I$. The second filter can be directly run if we already have $\hat{s}_I$, in other case, it must be computed using any of the methods in section 4, and then the second filter can be run, obtaining the estimates $\hat{s}_1,\cdots,\hat{s}_T$.

Example 1 (Continued)**.** *The matrices that define the two cascaded filters (12)-(13) for this example were obtained in section 2. We can proceed in several ways. For example, since* $\partial(\Omega_5(z))=2$, $\partial(e_3(z))=1$, $\partial(\Omega_2(z))=1$ *and* $\partial(d_4(z))=3$*, we can use the extended innovations algorithm or the respective forward and backward marginal models of* y_t *and* $\breve{y}_s$ *to get* $\hat{y}_F=\hat{y}_4=(.79,-.55)'$, $\hat{y}_I'=(\hat{y}_{-2}',\hat{y}_{-1}',\hat{y}_0')'$, $\hat{y}_{-2}=(.1,.063)'$, $\hat{y}_{-1}=(.28,.35)'$ *and* $\hat{y}_0=(.86,-.86)'$*. Then, using the left-right matrix geometric series we can compute* $\hat{v}_F'=(\hat{v}_4',\hat{v}_5')'$, $\hat{v}_4=(.094,.16,.91,.55)'$ *and* $\hat{v}_5=(.096,.45,.21,.75)'$*. Now we can run the first filter from* $s=1$ *to* $s=6$ *to obtain* $\hat{\breve{v}}_1=v_3=(.026,1.19,.31,2.09)'$, $\hat{v}_2=(-.24,-1.7,2.37,-2.46)'$, $\hat{v}_1=(.0096,3.47,-3.4,4.4)'$, $\hat{v}_0=(.41,-1.64,6.94,-2.43)'$, $\hat{v}_{-1}=(-.31,.72,-3.29,2.43)'$, $\hat{v}_{-2}=(-.18,.42,1.43,-.41)'$*. We compute* $\hat{s}_I=\hat{s}_0=(-.51,-.12)'$ *using the left-right matrix geometric series and run the second filter, obtaining the same* $\hat{s}_j$ *values as with the central difference equation.* ◇

5.3 Parallel implementation

$\hat{y}_I$ and $\hat{y}_F$ are needed, they can be computed using the forward and backward models of y_t or using the extended innovations algorithm. $\hat{s}_I^1$ and $\hat{s}_F^2$ are also needed and can be computed using any of the methods in section 4. Then both filters can be computed and their sum (after time-reversing $\hat{\breve{s}}_s^2$) provides the result.

Example 1 (Continued)**.** *The two parallel filters are*

$$\begin{pmatrix} 1-.95B & 0 \\ .17-.062B & 1 \end{pmatrix} s_t^1 = \begin{pmatrix} -.14+.15B-.030B^2 & .11B+.13B^2 \\ .02-.51B & -.016B \end{pmatrix} y_t$$

and

$$\begin{pmatrix} 1-.15B & 0 \\ .014 & 1 \end{pmatrix} \breve{s}_s^2 = \begin{pmatrix} .25-.46B & .37 \\ 0 & 0 \end{pmatrix} \breve{y}_s$$

The extended innovations algorithm gives $\hat{y}_I = \hat{y}_0 = (.86, -.86)'$, $\hat{y}_F = \hat{y}_4 = (.79, -.55)'$, $\hat{s}'_I = (\hat{s}'_{-1}, \hat{s}'_0)'$, $\hat{s}_{-1} = (-.029, -.043)'$, $\hat{s}_0 = (-.059, -.12)'$, *and* $\hat{s}_F = \hat{s}_4 = (-.19, .0027)'$. *Running the filters we obtain* $\hat{s}_1^1 = (-.12, -.39)'$, $\hat{s}_2^1 = (.072, -.55)'$, $\hat{s}_3^1 = (-.059, .29)'$, $\hat{\breve{s}}_1^2 = \hat{s}_3^2 = (.068, -.00096)'$, $\hat{\breve{s}}_2^2 = \hat{s}_2^2 = (-.63, .0089)'$ *and* $\hat{\breve{s}}_3^2 = \hat{s}_1^2 = (.72, -.01)'$. *Summing, we obtain the same result as with the other methods.* ◇

6 Forecasting, fixed point and fixed lag smoothing

Fixed point smoothing for a few fixed time points can be easily included in the framework of this paper. To do so, we simply compute the covariances of the S_t process with $y_{1:T}$ for t in the set of fixed time points and run the extended innovations algorithm.

Fixed lag smoothing can also be done, but efficiency is lost, due to the non-recursive nature of our procedure.

Forecasting or backcasting the s_t process for a fixed sample of y_t can also be easily included, in fact, the computation of the initial values of s_t when computing the second of the two cascaded filters already solved the backcasting problem.

Forecasting when the sample of the y_t process grows can also be done, but it faces again the loss in efficiency associated with our non-recursive procedure.

7 The ARIMA case

If the joint process of y_t and s_t needs some differencing to reach stationarity, we will face some additional problems. Assume that the joint model of s_t and y_t can be written in the multiplicative form $a(B)M(B)y_t = b(B)\epsilon_t$, where $M(z)$ is the differencing matrix polynomial that contains all the autoregressive roots with unit modulus. Let $\Phi(B)N(B)y_t = \Theta(B)u_t$ be the marginal model of y_t, also written in multiplicative form, where $N(z)$ contains all the roots of unit modulus. We define the stationary marginal process as $\psi_t = N(B)y_t$. We can forecast and backcast the ψ_t process, for example using the extended innovations algorithm. Then we can compute the forecasts and backcasts of y_t as explained in Aparicio-Perez (2011) and, using the Wiener-Kolmogorov filter equations, we can run the Wiener-Kolmogorov filter to obtain the estimations of the s_t values. Under some conditions that can be fulfilled even if the s_t and y_t processes are non-stationary, the relation between them is stationary (Gómez (2006)). The initial and final values of the filters that implement the Wiener-Kolmogorov filter can be obtained summing the corresponding left-right matrix geometric series.

An additional problem that can appear is the computation of $G_y(z)$ in (4): since we have pre-multiplied the MA part of the model by the adjoint of the AR part, the new MA part can contain several unit modulus roots, and the spectral factorization that is done to compute $G_y(z)$ may face some difficulties, like poor convergence properties.

8 Numerical computations

All the computations that are needed to implement the techniques explained in this paper can be done in an efficient and reliable way. We will briefly explain how to do so for some of them.

The computation of the adjoint of an n by n polynomial matrix $P(z)$ is sometimes done by transforming the polynomial matrix into an equivalent matrix pencil. We prefer, however, to use a more direct, and apparently new, method that consists of first transforming $P(z)$ into an upper triangular form $T(z)$. This can be done by premultiplication by a unimodular matrix $U(z)$, that is $T(z) = U(z)P(z)$, using the method explained in Henrion and Šebek (1999). The determinant of the polynomial matrix is simply obtained as the product of the diagonal elements of $T(z)$ divided by the (scalar) determinant of $U(z)$, and then the adjoint matrix $A(z)$ is easily obtained from the condition $A(z)P(z) = det(P(z))I_n$.

The transformation of a right into a left (or v.v.) matrix fraction description can also be done in a numerically reliable way, by solving a so-called minimal polynomial basis problem, see e.g. Chen(1984).

The problems of computing the autocovariances of a VARMA process and finding the spectral factorization of a vector moving average can both be solved by using the method explained in Henrion and Šebek (1998) to find the solution of a symmetric matrix polynomial equation.

The computation of the model that follows a filtered process can be done by transforming a right into a left matrix fraction and then doing a spectral factorization, see Aparicio-Pérez (2011). In this reference it is also explained how to compute the model that follows a time-reversed process.

9 Applications in official statistics

Filtering and smoothing is of central importance in time series analysis of macroeconomic time series. The non-recursive techniques exposed in this paper are well suited for this kind of applications, where the arrival of new information usually occurs at low frequency (daily or lower).

The classical fixed filters used in univariate time series signal extraction, like the Hodrick-Prescott filter, assume all a signal plus noise setting, with specific models for the signal and the noise. For this reason, the methods explained in this paper can be seen as a unifying framework for these filters.

But new univariate or multivariate filters can also be computed. As an illustration, we will now give some details of a particular structural bivariate model that could be used to extract the components of two observed economic indicators without using state space techniques.

Example 3. *Assume that two quarterly economic indicators y_{1t} and y_{2t} can be written each as the sum of a trend, a common seasonal component and a white noise, that is $y_{jt} = T_{jt} + S_t + e_{jt}$, $j \in \{1, 2\}$. The problem is to estimate the common seasonal component and the two trends.*

From all possible specifications for the components, we choose the following:

$$\begin{pmatrix} 1+B+B^2+B^3 & 0 & 0 & 0 & 0 \\ 0 & 1-B & 0 & 0 & 0 \\ 0 & 0 & (1-B)^2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} S_t \\ T_{1t} \\ T_{2t} \\ e_{1t} \\ e_{2t} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1+B & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \epsilon_{1t} \\ \epsilon_{2t} \\ \epsilon_{3t} \\ \epsilon_{4t} \\ \epsilon_{5t} \end{pmatrix}, \tag{28}$$

with ϵ_{jt} zero mean and unit variance uncorrelated gaussian variables, $j \in \{1, \cdots, 5\}$. First we compute the joint model of the five processes S_t, T_{1t}, T_{2t}, y_{1t} and y_{2t}. This is the model of the following filtered process:

$$\begin{pmatrix} S_t \\ T_{1t} \\ T_{2t} \\ y_{1t} \\ y_{2t} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} S_t \\ T_{1t} \\ T_{2t} \\ e_{1t} \\ e_{2t} \end{pmatrix}. \tag{29}$$

The result is

$$\begin{pmatrix} 1+B+B^2+B^3 & 0 & 0 & 0 & 0 \\ 0 & 1-B & 0 & 0 & 0 \\ 0 & 0 & (1-B)^2 & 0 & 0 \\ -1 & -1 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} S_t \\ T_{1t} \\ T_{2t} \\ y_{1t} \\ y_{2t} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1+B & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \epsilon_{1t} \\ \epsilon_{2t} \\ \epsilon_{3t} \\ \epsilon_{4t} \\ \epsilon_{5t} \end{pmatrix}. \tag{30}$$

The marginal model of y_{1t} *and* y_{2t} *is that of the filtered process*

$$\begin{pmatrix} y_{1t} \\ y_{2t} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} S_t \\ T_{1t} \\ T_{2t} \\ e_{1t} \\ e_{2t} \end{pmatrix}, \tag{31}$$

and we obtain (with two decimal places):

$$\begin{pmatrix} 1-B^4 & 0 \\ -(1-B)^2 & (1-B)^2 \end{pmatrix} \begin{pmatrix} y_{1t} \\ y_{2t} \end{pmatrix} = \begin{pmatrix} 1+.69B+.75^2+.67B^3 & -.25B-.25B^2-.34B^3 \\ -1+.96B+-.086B^2 & 1-.78B+.14B^2 \end{pmatrix} \begin{pmatrix} u_{1t} \\ u_{2t} \end{pmatrix},$$

with $cov(\begin{pmatrix} u_{1t} \\ u_{2t} \end{pmatrix} = \begin{pmatrix} 8.77 & 5.35 \\ 5.35 & 10.40 \end{pmatrix}$. *If, for example, we compute (5), we have*

$$d_{11}(B) = \begin{pmatrix} 1-3B+3B^2-B^3 & 0 & 0 \\ 0 & 1-B^2-B^4+B^6 & 0 \\ 0 & 0 & 1-B^4 \end{pmatrix}, d_{12}(B) = 0_{3\times 2},$$

$$\begin{bmatrix} d'_{21}(F) \\ d'_{22}(F) \end{bmatrix} = \begin{pmatrix} 1-3F+3F^2-F^3 & 1-3F+3F^2-F^3 \\ 1-F^2-F^4+F^6 & 0 \\ 0 & 1-F^4 \\ 1-2F+F^2-F^4+2F^5-F^6 & 0 \\ 0 & 1-2F+F^2-F^4+2F^5-F^6 \end{pmatrix},$$

$\Omega_{11}(B) = 2.96-4.33B+1.54B^2-.59B^3-.93B^4+1.35B^5$, $\Omega_{12}(B) = -.66B+.65B^2-.22B^3+1.15B^4-.92B^5$, $\Omega_{21}(B) = 1.81-2.89B+1.54B^2-.59B^3+.22B^4-.083B^5$ *and* $\Omega_{22}(B) = 2.67-2.75B+1.02B^2-.22B^3-1.52B^4+1.17B^5-.37B^6$ ◇

This example suggests a new general methodology that could be used to treat multivariate structural time series models using polynomial methods. After writing a structural model in polynomial form, an optimization routine would be used to estimate its parameters, and then the implementation of the Wiener-Kolmogorov filter proposed in this paper could be used to estimate the unobserved components.

10 Conclusions

This paper has shown how to compute a multivariate Wiener-Kolmogorov filter without using state-space techniques. The procedure is computationally efficient and numerically reliable. Some applications in official statistics have been outlined.

Many of the techniques that are proposed in the paper can be optimized taking into account the Kronecker indices of the processes that are involved. This has not been done in the paper in order to keep the exposition so simple as possible. As an example, in section

4.2 it may be possible, for $k > k_1$ and depending on the Kronecker indices of the backward process (25), to backcast the values of y_k from less than all the components of $y_{k-k_1}, \cdots, y_{k-1}$, but the inclusion of the details on how to do so would make the paper more difficult to read, without altering the essential ideas. See sections 1 and 2 in Hannan and Deistler (1988) for a definition of the Kronecker indices and some applications of polynomial matrices and state-space methods to discrete time linear stochastic systems theory.

APPENDIX A: Right and left matrix fraction descriptions

Leaving aside the algebraic properties, we provide some basic definitions that involve polynomial matrices and a result on right and left matrix fraction descriptions.

Definition A1.
a) A $s \times m$ polynomial matrix is an $s \times m$ array that has as elements polynomials in a scalar variable.
b) A polynomial matrix is said to be unimodular if its determinant is a constant.
c) Let $P(z)$ and $Q(z)$ be two $s \times m$ polynomial matrices. If $P(z) = P_1(z) \cdot M(z)$ ($P(z) = L(z) \cdot P_1(z)$) and $Q(z) = Q_1(z) \cdot M(z)$ ($Q(z) = L(z) \cdot Q_1(z)$) with $M(z)$ ($L(z)$) a polynomial matrix, $M(z)$ ($L(z)$) is called a common right divisor (common left divisor) of $P(z)$ and $Q(z)$ and $P(z)$ and $Q(z)$ are said to be multiples of $P_1(z)$ and $Q_1(z)$ respectively.
d)If $M(z)$ ($L(z)$) is a multiple of any other common right (left) divisor of $P(z)$ and $Q(z)$, it is called a greatest common right (left) divisor.

Definition A2.
a) A $s \times m$ rational transfer function $T(z)$ is a $s \times m$ array that has as elements polynomial quotients.
b) A right coprime fraction (r.c.f) or right coprime matrix fraction description, of $T(z)$ is a pair of polynomial matrices, $(N_r(z), D_r(z))$, of orders $s \times m$ and $m \times m$ respectively such that:
(i) $D_r(z)$ is non-singular.
(ii) $T(z) = N_r(z)D_r(z)^{-1}$.
(iii) $(N_r(z), D_r(z))$ is right-coprime, that is, all its greatest common right divisors are unimodular matrices.
c) A left coprime fraction (l.c.f) or left coprime matrix fraction description, of $T(z)$ is a pair of polynomial matrices, $(N_l(z), D_l(z))$, of orders $s \times m$ and $s \times s$ respectively such that:
(i) $D_l(z)$ is non-singular.
(ii) $T(z) = D_l(z)^{-1}N_l(z)$.
(iii) $(N_l(z), D_l(z))$ is left-coprime, that is, all its greatest common left divisors are unimodular matrices.

Theorem A1. *Given a $s \times m$ rational transfer function $T(z)$, it can always be expressed as a r.c.f. or l.c.f. $T(z) = D_l(z)^{-1}N_l(z) = N_r(z)D_r(z)^{-1}$*

For a proof see e.g. Kailath(1980), sections 6.3 and 6.5
We provide an example of a 2×1 transfer function expressed as a r.c.f and a l.c.f.

$$T(z) = \begin{pmatrix} z(z-1)(z+2) \\ z+1 \end{pmatrix} \cdot ((z+1)(z-1))^{-1} = \begin{pmatrix} z+1 & z-1 \\ 0 & (z-1)^2 \end{pmatrix}^{-1} \cdot \begin{pmatrix} (z+1)^2 \\ z-1 \end{pmatrix}$$

References

[1] Ahlén and Sternad (1991), *Wiener Filter Design Using Polynomial Equations*, IEEE Transactions on Signal Processing, vol. 39, no. 11, 2387–2399.

[2] Aparicio-Pérez, F. (2010): *Polynomial Methods in Time Series Analysis*, presented at COMPSTAT 2010; Paris, France, Springer-Verlag.

[3] Aparicio-Pérez, F. (2011): *Matrix Polynomial Computations with Applications in Time Series Analysis*, Working paper, INE, Madrid, Spain, (forthcoming).

[4] Burman, J.P. (1980), *Seasonal Adjustment by Signal Extraction*, J.R. Statist. Soc. A, vol. 143, part 3, 321–337.

[5] Brockwell, P.J., and Davis, R.A. (1991), *Time Series Analysis, Forecasting and Control* San Francisco: Holden–Day.

[6] Caines, P.E. (1988), *Linear Stochastic Systems*, New York: John Wiley.

[7] Chen, C.T. (1984), *Linear System Theory and Design*, Holt, Rinehart and Winston.

[8] Gómez, V. (2006), *Wiener-Kolmogorov Filtering and Smoothing for Multivariate Series with State-Space Structure*, Journal of Time Series Analysis, vol. 28, No. 3, 361–385.

[9] Hannan, E.J. and Deistler, M. (1988), *The Statistical Theory of Linear Systems*, New York: John Wiley.

[10] Henrion, D. and Sebek, M. (1998), *Efficient Numerical Method for the Discrete-time Symmetric Matrix Polynomial Equation*, IEEE Proceedings, Control Theory and applications. 145 (5), 143–148.

[11] Henrion, D. and Sebek, M. (1999), *Reliable Numerical Methods for Polynomial Matrix Triangularization*, IEEE. Transactions on Automatic Control, vol. 44, no. 3, pp 497–508.

[12] Kailath, T. (1980), *Linear Systems*, Prentice-Hall: Englewood Cliffs, NJ.

[13] Kalman, R.E. (1960), *New Results in Linear Filtering and Prediction Theory*, J. of Basic Eng., Trans. ASME, Series D, Vol. 82, No. 1, pp 35-45.

[14] Kenney, C. and Laub, A.J. (1988), *Controllability and Stability Radii for Companion Form Systems*, Math. Contr. Signals and Syst. vol. 1, pp. 239-256. Reprinted in Patel, R.V., Laub, A.J. and Van Dooren, P.M. (eds.) (1994) *Numerical Linear Algebra Techniques for Systems and Control*, IEEE Press: New York.

[15] Kolmogorov, A.N. (1941), *Interpolation and Extrapolation*, Bulletin de l'Academie des Sciences de U.S.S.R., Series Mathematics 5, pp 3-14.

[16] Lancaster, P. and Tismenetsky, M. (1985), *The theory of matrices, second edition, with applications*, San Diego: Academic Press.

[17] Roberts, A.P. and Newmann, M.M. (1988), *Polynomial Approach to Wiener Filtering*, International Journal of Control, vol 47, pp 681–696.

[18] Wiener, N. (1949), *Extrapolation, Interpolation and Smoothing of Stationary Time Series*, *The Mit Press, Cambridge, MA*. Originally published in 1942 as a classified document.