CHEATING AT CHOLESKY WITH THE KALMAN FILTER

by

R. L. Eubank and Suojin Wang

Department of Statistics

Texas A&M University

College Station, TX 77843

Abstract

This is an expository article that develops the Kalman filter from a Cholesky factorization perspective. In particular, the Kalman filter is shown to be a modification of the Cholesky factorization that arises from efficient use of the intrinsic structure in state-space models.

Keywords: Cholesky decomposition, Kalman filter, order n algorithm.

1. Introduction. The Kalman filter (KF) provides an efficient algorithm for computing conditional means, predictions and other related quantities in state-space models. The KF's origins lie in the engineering literature (Kalman, 1960, and Kalman and Bucy, 1961), but it has become of increasing interest to statisticians due to statistics-friendly articles such as Meinhold and Singpurwalla (1983). One can now find statistical applications of the KF in such diverse areas as time series and nonparametric smoothing.

In spite of increased statistical awareness, the properties and uses of the KF can still seem rather mysterious upon first encounter. This is particularly true in comparison to classical computing methods (such as the Cholesky decomposition) which are more familiar to statisticians. While it is well known (e.g., Kohn and Ansley, 1989, p. 77) that the KF is effectively computing a Cholesky factorization of a variance-covariance matrix, simple expository explanations of this fact are not readily found. In addition, for problems of dimension n, the KF carries out its computations in only order n steps while a standard Cholesky approach requires computations of order n^2. This raises questions as to how the KF can be so efficient if it is also (at least implicitly) carrying out a Cholesky factorization.

One way to understand how the KF works is to examine the Cholesky factorization of the relevant variance-covariance matrix for a state-space model. We do this in Section 3 where we

see that one can easily obtain closed form expressions for the components of the decomposition. Not surprisingly, these components have a very specific structure that is easy to exploit in a way that converts the standard Cholesky approach from an order n^2 to an order n algorithm. In Section 4 this modified algorithm is shown to be identical to the KF with the forward and back-substitution steps of the Cholesky algorithm producing, respectively, the forward and backward (or smoothing) steps for the KF.

As previously noted, connections between the KF and Cholesky method have been known for decades. Thus, we make no claims of novelty in what follows. Instead, a goal of this paper is to give a simple, one source, expository treatment where the KF can be easily recognized as a clever version of the Cholesky method. Our hope is that this will make the KF seem less mysterious to those with classical statistical computing backgrounds and thereby enhance its use in practice.

The results in this paper also have pedagogical implications. One can, for example, use the developments in Section 3 to teach the KF algorithm entirely from a Cholesky perspective. When teaching the Cholesky method in statistical computing curricula it is standard practice to treat the case of band-limited matrices to exemplify how the algorithm can be modified to be more efficient in the presence of special structure. Our work in Section 3 provides another example of an efficient Cholesky modification that can be presented to students with the added benefit of giving a forum for introduction of the KF outside of the usual time series setting.

2. Background. In this section we develop a simple framework where the connection between the KF and Cholesky methods is easy to explain. We begin with a discussion of Cholesky factorization in the context of a signal-plus-noise model.

Suppose that we have responses $y(1), \ldots, y(n)$ following the model

$$y(t) = f(t) + e(t), \quad t = 1, \ldots, n,$$

where $e(1), \ldots, e(n)$ are iid normal random variables with zero mean and variance σ^2 and $f(1), \ldots, f(n)$ are normal random variables that are independent of $e(1), \ldots, e(n)$, have zero means and covariances

$$\mathrm{cov}\{f(s), f(t)\} = \sigma^2 Q(s,t), \quad s, t = 1, \ldots, n.$$

In matrix-vector form we can write the model above as

$$\mathbf{y} = \mathbf{f} + \mathbf{e} \tag{1}$$

with $E(\mathbf{e}) = 0$, $\mathrm{cov}(\mathbf{e}) = \sigma^2 I, E(\mathbf{f}) = 0$ and $\mathrm{cov}(\mathbf{f}) = \sigma^2 Q$, for $Q = \{Q(s,t)\}_{s,t=1,n}$.

A typical estimation problem for model (1) is to compute the mean-squared error optimal predictor of the signal vector $\mathbf{f}$. More precisely, we want to evaluate the conditional expectation

$$\widehat{\mathbf{f}} = E[\mathbf{f}|\mathbf{y}] = E[\mathbf{y} - \mathbf{e}|\mathbf{y}] = \mathbf{y} - \widehat{\mathbf{e}} \tag{2}$$

for

$$\widehat{\mathbf{e}} = E[\mathbf{e}|\mathbf{y}].$$

Due to our normality and independence assumptions, standard multivariate analysis results give

$$\widehat{\mathbf{e}} = \mathrm{cov}(\mathbf{e}, \mathbf{y})\{\mathrm{cov}(\mathbf{y})\}^{-1}\mathbf{y} = (Q + I)^{-1}\mathbf{y}. \tag{3}$$

To compute $\widehat{\mathbf{e}}$ one can therefore employ the Cholesky factorization

$$Q + I = LRL^T \tag{4}$$

for L an $n \times n$ lower triangular matrix with ones on the diagonal and R a diagonal matrix with diagonal entries $R(1), \ldots, R(n)$, i.e.,

$$R = \mathrm{diag}(R(1), \ldots, R(n)).$$

Using (4) in (3) gives

$$L^T\widehat{\mathbf{e}} = R^{-1}L^{-1}\mathbf{y}, \tag{5}$$

which has the consequence that $\widehat{\mathbf{e}}$ can be obtained by first computing $R^{-1}L^{-1}\mathbf{y}$ and then back-substituting using the elements of the upper-triangular matrix L^T to obtain, successively, $\widehat{e}(n), \widehat{e}(n-1), \ldots, \widehat{e}(1)$.

To obtain

$$\boldsymbol{\varepsilon} = (\varepsilon(1), \ldots, \varepsilon(n))^T = L^{-1}\mathbf{y}$$

and R one employs the Gramm-Schmidt process whereby

$$\varepsilon(1) = y(1),$$
$$R(1) = \mathrm{var}\{y(1)\}/\sigma^2,$$

and, for $t = 2, \ldots, n$,

$$\begin{aligned}\varepsilon(t) &= y(t) - E[y(t)|\varepsilon(1), \ldots, \varepsilon(t-1)] \\ &= y(t) - E[y(t)|y(1), \ldots, y(t-1)] \\ &= y(t) - \sum_{j=1}^{t-1} \ell_{tj}\varepsilon(j) \end{aligned} \tag{6}$$

with

$$R(t) = \text{var}\{\varepsilon(t)\}/\sigma^2 \tag{7}$$

and

$$\ell_{tj} = \frac{\text{cov}\{y(t), \varepsilon(j)\}}{\sigma^2 R(j)}, \quad j = 1, \ldots, t-1. \tag{8}$$

The equivalence of the Gramm-Schmidt and Cholesky algorithms comes from observing that the transformation from $\mathbf{y}$ to $\boldsymbol{\varepsilon}$ is linear, i.e., $\boldsymbol{\varepsilon} = T\mathbf{y}$ for some matrix T. Since $\text{cov}(\boldsymbol{\varepsilon}) = \sigma^2 R$, and the Cholesky decomposition is unique, it follows that $T = L^{-1}$. But $\boldsymbol{\varepsilon} = L^{-1}\mathbf{y}$ in conjuction with (6) entails that the below diagonal elements of L are given by (8).

In general the Cholesky method for computing $\widehat{\mathbf{e}}$ will require order n^2 calculations. However, there are instances when the number of computations can be substantially reduced. The standard example of this is when $Q + I$ is band-limited in which case the Cholesky method can be modified to efficiently compute $\widehat{\mathbf{e}}$ in only order n operations. (See, e.g., Franklin 1968, Section 7.4.) In the next section we will demonstrate that state-space models provide another example of where special structure allows for the construction of a modified, order n, Cholesky algorithm for computing $\widehat{\mathbf{e}}$.

3. Cholesky in State-Space. We now specialize the discussion of the previous section to the context of state-space models. Specifically, we now assume that $f(t)$ in (1) is given by

$$f(t) = \mathbf{h}^T(t)\mathbf{x}(t) \tag{9}$$

for $\mathbf{h}(t)$, $t = 1, \ldots, n$, known p-vectors and $\mathbf{x}(t), t = 1, \ldots, n$, random "state" vectors which satisfy

$$\mathbf{x}(t+1) = F(t)\mathbf{x}(t) + \mathbf{u}(t) \tag{10}$$

for $F(t)$, $t = 0, \ldots, n-1$, known $p \times p$ matrices and $\mathbf{u}(0), \ldots, \mathbf{u}(n-1)$ independent, zero mean normal random vectors with variance-covariance matrices

$$\text{cov}\{\mathbf{u}(t)\} = \sigma^2 Q(t), \quad t = 0, \ldots, n-1, \tag{11}$$

that are also independent of the random vector $\mathbf{e}$ in (1). To start the recursion (10) we also need an initial state vector $\mathbf{x}(0)$ which is often assumed to be a zero mean normal random vector that is independent of the $\mathbf{u}(t)$, $t = 0, \ldots, n-1$, and $\mathbf{e}$. The variance-covariance matrix for $\mathbf{x}(0)$ will be denoted by

$$\mathrm{cov}\{\mathbf{x}(0)\} = \sigma^2 S(0|0). \tag{12}$$

When (9)–(12) are in effect, model (1) becomes a state-space model.

In this section we will use the state-space structure to develop expressions for L and R that will facilitate creation of an efficient Cholesky algorithm for computing $\widehat{\mathbf{e}}$. This involves the use of several recursive relations that are direct consequences of the state-space model.

Let

$$S(t|t-1) = \mathrm{cov}\{\mathbf{x}(t)|\varepsilon(1), \ldots, \varepsilon(t-1)\}/\sigma^2 \tag{13}$$

with the initializing convention that

$$S(1|0) \equiv \mathrm{cov}(\mathbf{x}(1)) = F(0)S(0|0)F^T(0) + Q(0). \tag{14}$$

Then, one finds that

$$\begin{aligned} S(t|t-1) = Q(t-1) + F(t-1)[&S(t-1|t-2) \\ &- S(t-1|t-2)\frac{\mathbf{h}(t-1)\mathbf{h}^T(t-1)}{R(t-1)}S(t-1|t-2)]F^T(t-1), t = 2, \ldots, n, \end{aligned} \tag{15}$$

$$\mathrm{cov}\{\mathbf{x}(t), \varepsilon(t)\} = \sigma^2 S(t|t-1)\mathbf{h}(t), \quad t = 1, \ldots, n, \tag{16}$$

and $R(t)$ in (7) becomes

$$R(t) = \mathbf{h}^T(t)S(t|t-1)\mathbf{h}(t) + 1, \quad t = 1, \ldots, n. \tag{17}$$

Identities (15)–(17) have immediate implications for recursive computation of L. Specifically, let

$$\mathbf{c}_{tj} = \mathrm{cov}\{\mathbf{x}(t), \varepsilon(j)\}/\sigma^2 R(j).$$

Then, from (8) and (9) we see that the tth step of the Cholesky forward recursion produces $\ell_{tt} = 1$ and

$$\begin{aligned} \ell_{tj} &= \mathrm{cov}\{y(t), \varepsilon(j)\}/\sigma^2 R(j) \\ &= \mathbf{h}^T(t)\mathbf{c}_{tj} \end{aligned}$$

for $j = 1, \ldots, t-1$. However, from (10),

$$\mathbf{c}_{tj} = F(t-1)\mathbf{c}_{(t-1)j} \tag{18}$$

with the quantities $\mathbf{c}_{(t-1)j}, j = 1, \ldots, t-2$, being available from the $(t-1)$st step of the recursion and

$$\mathbf{c}_{(t-1)(t-1)} = S(t-1|t-2)\mathbf{h}(t-1)/R(t-1) \tag{19}$$

due to (16). This reveals that the matrix L can be computed row-by-row in order n^2 computations by updating $\mathbf{c}_{(t-1)j}, j = 1, \ldots, t-2$, on the tth step along with one new computation for $\mathbf{c}_{t(t-1)}$. The basic computing scheme is illustrated in Figure 1.

$$\begin{bmatrix} \mathbf{c}_{(t-1)1} \\ \mathbf{c}_{(t-1)2} \\ \vdots \\ \mathbf{c}_{(t-1)(t-2)} \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{c}_{t1} = F(t-1)\mathbf{c}_{(t-1)1} \\ \mathbf{c}_{t2} = F(t-1)\mathbf{c}_{(t-1)2} \\ \vdots \\ \mathbf{c}_{t(t-2)} = F(t-1)\mathbf{c}_{(t-1)(t-2)} \\ \mathbf{c}_{t(t-1)} = F(t-1)\dfrac{S(t-1|t-2)\mathbf{h}(t-1)}{R(t-1)} \end{bmatrix} \rightarrow \begin{bmatrix} \ell_{t1} = \mathbf{h}^T(t)\mathbf{c}_{t1} \\ \ell_{t2} = \mathbf{h}^T(t)\mathbf{c}_{t2} \\ \vdots \\ \ell_{t(t-2)} = \mathbf{h}^T(t)\mathbf{c}_{t(t-2)} \\ \ell_{t(t-1)} = \mathbf{h}^T(t)\mathbf{c}_{t(t-1)} \end{bmatrix}$$

Figure 1. Forward Row-By-Row Recursion for L

The vector $\boldsymbol{\varepsilon} = L^{-1}\mathbf{y}$ can, however, be computed without explicitly evaluating either L or L^{-1}. To see this let $\varepsilon(1) = y(1)$ and $\mathbf{a}_0 = \mathbf{0}_p$, for $\mathbf{0}_p$ the p-dimensional zero vector. Then, from (6), (8) and (18)–(19), for $t = 2, \ldots, n$, we have

$$\varepsilon(t) = y(t) - \mathbf{h}^T(t)F(t-1)\mathbf{a}_{t-1} \tag{20}$$

with

$$\mathbf{a}_{t-1} = \sum_{j=1}^{t-1} \mathbf{c}_{(t-1)j}\varepsilon(j) \tag{21}$$

$$= \mathbf{c}_{(t-1)(t-1)}\varepsilon(t-1) + F(t-2)\mathbf{a}_{t-2}. \tag{22}$$

The combination of (20) and (22) (along with (15) and (17)) provide an order n algorithm for computing $\boldsymbol{\varepsilon}$. The key ingredient in the recursion is (22) which allows us to replace the time consuming summation implicit in (21) with an efficient, "one-step" updating process.

It now remains only to efficiently back-solve the system (5). For this purpose, we need a parallel of the recursion (18)–(19) that is applicable to the columns, rather than the rows, of L. This is obtained by using the state-space structure along with (16) to see that for $t > j$

$$\begin{aligned}\ell_{tj} &= \mathrm{cov}\{y(t), \varepsilon(j)\}/\sigma^2 R(j) \\ &= \mathbf{h}^T(t)F(t-1)\cdots F(j)\mathrm{cov}\{\mathbf{x}(j), \varepsilon(j)\}/\sigma^2 R(j) \\ &= \mathbf{b}_{tj}^T S(j|j-1)\mathbf{h}(j)/R(j) \end{aligned} \tag{23}$$

with

$$\mathbf{b}_{tj} = F^T(j)\mathbf{b}_{t(j+1)} \tag{24}$$

for $t = j+2, \ldots, n$ and

$$\mathbf{b}_{(j+1)j} = F^T(j)\mathbf{h}(j+1). \tag{25}$$

Equations (23)–(25) provide an alternative to (18)–(19) that allows us to construct L column-by-column in order n^2 steps as illustrated diagrammatically in Figure 2.

$$\begin{bmatrix} \ell_{(j-1)(j-1)} = 1 \\ \ell_{j(j-1)} = \mathbf{b}_{j(j-1)}^T \dfrac{S(j-1|j-2)\mathbf{h}(j-1)}{R(j-1)} \\ \ell_{(j+1)(j-1)} = \mathbf{b}_{(j+1)(j-1)}^T \dfrac{S(j-1|j-2)\mathbf{h}(j-1)}{R(j-1)} \\ \vdots \\ \ell_{n(j-1)} = \mathbf{b}_{n(j-1)}^T \dfrac{S(j-1|j-2)\mathbf{h}(j-1)}{R(j-1)} \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{b}_{j(j-1)} = F(j-1)\mathbf{h}(j) \\ \mathbf{b}_{(j+1)(j-1)} = F(j-1)\mathbf{b}_{(j+1)j} \\ \vdots \\ \mathbf{b}_{n(j-1)} = F(j-1)\mathbf{b}_{nj} \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{b}_{(j+1)j} \\ \vdots \\ \mathbf{b}_{nj} \end{bmatrix}$$

Figure 2. Backward Column-By-Column Recursion for L

Now define $\tilde{\mathbf{a}}_n = \mathbf{0}_p$ and observe that

$$\widehat{e}(n) = \varepsilon(n)/R(n). \tag{26}$$

In general, using (23)–(25) for $j = n, \ldots, 2$, we have

$$\widehat{e}(j-1) = \frac{\varepsilon(j-1)}{R(j-1)} - \sum_{t=j}^{n} \widehat{e}(t)\ell_{t(j-1)} \tag{27}$$

$$= \frac{\varepsilon(j-1)}{R(j-1)} - \tilde{\mathbf{a}}_{j-1}^T \frac{S(j-1|j-2)\mathbf{h}(j-1)}{R(j-1)}, \tag{28}$$

where

$$\tilde{\mathbf{a}}_{j-1} = \widehat{e}(j)F^T(j-1)\mathbf{h}(j) + F^T(j-1)\tilde{\mathbf{a}}_j. \tag{29}$$

The algorithm obtained from (26), (28) and (29) is clearly of order n. The key component is the recursion (29) that allows us to avoid actually doing the summation in (27).

To summarize, we have developed a modified Cholesky algorithm that can be used to compute $\widehat{\mathbf{e}}$ in (3) in order n operations under the state-space model (9)–(12). On the forward pass we evaluate $\boldsymbol{\varepsilon} = L^{-1}\mathbf{y}$ using (14), (15) and (17) in conjunction with (20) and (22). Then, to obtain $\widehat{\mathbf{e}}$ we back-solve $L^T\widehat{\mathbf{e}} = R^{-1}\boldsymbol{\varepsilon}$ using (26), (28) and (29). In the next section we will demonstrate that our modified Cholesky algorithm is precisely the KF algorithm.

4. The Kalman Filter. Perhaps the most common way to compute $\boldsymbol{\varepsilon}$ and $\widehat{\mathbf{e}}$ for state-space models is by use of the KF. In particular, the KF computes the $\boldsymbol{\varepsilon}$ vector using the fact that

$$\begin{aligned}\varepsilon(t) &= y(t) - E[y(t)|y(1),\ldots,y(t-1)] \\ &= y(t) - \mathbf{h}^T(t)\widehat{\mathbf{x}}(t|t-1),\end{aligned}$$

where $\widehat{\mathbf{x}}(t|t-1)$ is the conditional expectation of the state vector $\mathbf{x}(t)$ given the observed responses up to time $t-1$. It then uses an updating process that allows one to iteratively compute

$$\widehat{\mathbf{x}}(t|t) = E[\mathbf{x}(t)|y(1),\ldots,y(t)]$$

from $\widehat{\mathbf{x}}(t|t-1)$ and $\varepsilon(t)$ and combines this with the relation $\widehat{\mathbf{x}}(t+1|t) = F(t)\widehat{\mathbf{x}}(t|t)$ to proceed to the computation of $\varepsilon(t+1)$. The process begins with $\widehat{\mathbf{x}}(0|0) = 0$.

Detailed discussions and derivations of the forward Kalman recursions for $\boldsymbol{\varepsilon}$ can be found, for example, in Priestley (1981, Section 10.6) and Meinhold and Singpurwalla (1983). We merely give the form of the recursion here so that it can be compared to the results in Section 3.

The Kalman filter employs (14)–(17) to obtain

$$\varepsilon(t) = y(t) - \mathbf{h}^T(t)F(t-1)\widehat{\mathbf{x}}(t-1|t-1) \tag{30}$$

using the relation

$$\begin{aligned}\widehat{\mathbf{x}}(t-1|t-1) &= F(t-2)\widehat{\mathbf{x}}(t-2|t-2) \\ &\quad + \mathbf{k}(t-1)\varepsilon(t-1),\end{aligned} \tag{31}$$

where $\mathbf{k}(t-1)$ is the Kalman gain vector

$$\mathbf{k}(t-1) = \frac{S(t-1|t-2)\mathbf{h}(t-1)}{R(t-1)}. \tag{32}$$

Comparing (30)–(32) to (20) and (22), we see that $\mathbf{a}_{(t-1)} = \widehat{\mathbf{x}}(t-1|t-1)$ and $\mathbf{k}(t-1) = \mathbf{c}_{(t-1)(t-1)}$. Thus, the KF recursion for computing $\boldsymbol{\varepsilon}$ is identical to our modified forward Cholesky algorithm in Section 3.

At the end of the forward recursion for the KF we have computed the vector $\boldsymbol{\varepsilon}$, the variance/covariance factors (13) and (17) and an estimator of $\mathbf{f}$ in (1) of the form

$$\tilde{\mathbf{f}} = \begin{bmatrix} \mathbf{h}^T(1)\widehat{\mathbf{x}}(1|1) \\ \mathbf{h}^T(2)\widehat{\mathbf{x}}(2|2) \\ \vdots \\ \mathbf{h}^T(n)\widehat{\mathbf{x}}(n|n) \end{bmatrix} = \begin{bmatrix} E[f(1)|y(1)] \\ E[f(2)|y(1), y(2)] \\ \vdots \\ E[f(n)|y(1), \ldots, y(n)] \end{bmatrix}.$$

This estimator may be satisfactory for some purposes such as certain time series applications. However, in many instances one would prefer the estimator (2) which requires us to replace $E[f(t)|y(1), \ldots, y(t-1)]$ with $\widehat{f}(t) = E[f(t)|y(1), \ldots, y(n)]$. This is called the smoothing part of the KF and requires a further backward recursion to update the elements of $\tilde{\mathbf{f}}$ to include "new" y information.

Perhaps the most efficient algorithm for conducting the smoothing step of the KF was developed relatively recently by Kohn and Ansley (1989). We will now show that their algorithm is identical to the back-substitution phase of the modified Cholesky algorithm in Section 3.

Let $M(j) = F(j) - F(j)\mathbf{k}(j)\mathbf{h}^T(j)$, define $\mathbf{d}(n) = \mathbf{0}_p$, and let $\widehat{e}(n)$ be as in (26). Then, the Kohn and Ansley (1989) smoothing algorithm proceeds by taking

$$\mathbf{d}(j-1) = -\mathbf{h}(j)\varepsilon(j)/R(j) + M^T(j)\mathbf{d}(j)$$

and

$$\widehat{e}(j-1) = \varepsilon(j-1)/R(j-1) + \mathbf{d}^T(j-1)F(j-1)\mathbf{k}(j-1)$$

for $j = n, \ldots, 2$. To establish that this is the same as (28)–(29) it suffices, in view of (32), to show that $F^T(j-1)\mathbf{d}(j-1) = -\tilde{\mathbf{a}}_{j-1}$ for $j = n+1, \ldots, 2$. This follows easily by induction.

5. An example. We conclude with an example that provides a simple illustration of some of the ideas in the previous sections. For this purpose take $p = 1$ in (9)-(10) which reduces all the previous matrices and vectors to scalars. Accordingly, we drop the boldface font for vectors and

further simplify matters by assuming that $F(0) = \cdots = F(n-1) = h(1) = \cdots = h(n) = 1$. In this instance the KF or modified Cholesky forward recursion (30)–(32) or (20)and (22) begins with the initial step

$$S(1|0) = S(0|0) + Q(0),$$

$$R(1) = S(1|0) + 1,$$

$$\varepsilon(1) = y(1),$$

and

$$\widehat{x}(1|1) = a_1 = k(1)\varepsilon(1)$$

for $k(1) = c_{11}$ the first Kalman gain factor given by

$$k(1) = \frac{S(1|0)}{R(1)}.$$

On the tth step ($t = 2, \ldots, n$) relations (15) and (17) reduce to

$$S(t|t-1) = Q(t-1) + S(t-1|t-2) - \frac{S^2(t-1|t-2)}{R(t-1)}$$

and

$$R(t) = S(t|t-1) + 1.$$

Then, from (20) and (30), we have

$$\begin{aligned}\varepsilon(t) &= y(t) - a_{t-1} \\ &= y(t) - \widehat{x}(t|t-1),\end{aligned}$$

with

$$\begin{aligned}\widehat{x}(t|t) &= a_{t-1} + c_{tt}\varepsilon(t) \\ &= \widehat{x}(t-1|t-1) + k(t)\varepsilon(t)\end{aligned}$$

since, from (19) and (32),

$$k(t) = \frac{S(t|t-1)}{R(t)} = c_{tt}.$$

The quantities c_{tj} and b_{tj} in (18)–(19) and (24)–(25) are now seen to be $c_{tj} = k(j), j = 1, \ldots, t-1$ and $b_{tj} = 1, t = j+1, \ldots, n$. Thus, the matrix L in (4) has the simple representation

$$L = \begin{bmatrix} 1 & & & \ldots & 0 \\ k(1) & 1 & & \ldots & 0 \\ k(1) & k(2) & 1 & \ldots & 0 \\ k(1) & k(2) & k(3) & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ k(1) & k(2) & k(3) & \ldots & 1 \end{bmatrix}.$$

The forward, or Gramm-Schmidt, portion of the algorithm takes the form

$$\begin{aligned}\varepsilon(t) &= y(t) - \sum_{j=1}^{t-1} \ell_{tj}\varepsilon(j) \\ &= y(t) - \sum_{j=1}^{t-1} k(j)\varepsilon(j) \\ &= y(t) - a_{t-1},\end{aligned}$$

where a_{t-1} is computed recursively from (22) by $a_{t-1} = \varepsilon(t-1)k(t-1) + a_{t-2}$. Thus, efficient computation is obtained by updating the partial sums at each step (using only one multiplication and an addition) rather than direct computation of $\sum_{j=1}^{t-1} k(j)\varepsilon(j)$ which would require some $2(t-1)$ operations.

For the backward recursion we have $\widehat{e}(n) = \varepsilon(n)/R(n)$ and, for $t = n-1, \ldots, 1$,

$$\begin{aligned}\widehat{e}(t) &= \frac{\varepsilon(t)}{R(t)} - \sum_{j=t+1}^{n} \ell_{jt}\widehat{e}(j) \\ &= \frac{\varepsilon(t)}{R(t)} - \sum_{j=t+1}^{n} \widehat{e}(j)k(t) \\ &= \frac{\varepsilon(t)}{R(t)} - \tilde{a}_t k(t)\end{aligned}$$

with $\tilde{a}_t$ updated from (29) using

$$\tilde{a}_t = \widehat{e}(t) + \tilde{a}_{t+1}.$$

The conclusion is similar to the forward case in that efficiency requires accumulation and updating of the partial sums $\sum_{j=t+1}^{n} \widehat{e}(j)$ rather than recomputation of the entire sum on each step.

This example corresponds to a simple setting where one can easily see how the modified Cholesky/KF algorithm uses the state-space structure to produce an efficient, order n, computational method. However, the difference between this example and the general case is more notational than conceptual. The key element in both scenarios is the one-step updating of sums in the forward and backward recursions that is made possible by the state-space framework.

Acknowledgments. Eubank's research was supported in part by the National Science Foundation. Wang's research was supported in part by the Texas Advanced Research Program, the National Cancer Institute (CA57030) and the TAMU Center for Environmental and Rural Health.

References

Franklin, J.N. (1968). *Matrix Theory.* Prentice Hall: New Jersey.

Kalman, R. E. (1960). "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, 82, 34–45.

Kalman, R. E. and Bucy, R. S. (1961). "New Results in Linear Filtering and Prediction Theory," *Journal of Basic Engineering*, 85, 95–108.

Kohn, R. and Ansley, C. F. (1989). "A Fast Algorithm for Signal Extraction, Influence and Cross-Validation in State-Space Models," *Biometrika*, 76, 65–79.

Meinhold, R. J. and Singpurwalla, N. D. (1983). "Understanding the Kalman Filter," *American Statistician*, 37, 123–127.

Priestley, M.B. (1981). *Spectral Analysis and Time Series.* Academic Press: New York.