SIAM J. NUMER. ANAL.
Vol. 14, No. 2, April 1977

LEAST SQUARES ESTIMATION OF DISCRETE LINEAR DYNAMIC SYSTEMS USING ORTHOGONAL TRANSFORMATIONS*

C. C. PAIGE† AND M. A. SAUNDERS‡

Abstract. Kalman [9] introduced a method for estimating the state of a discrete linear dynamic system subject to noise. His method is fast but has poor numerical properties. Duncan and Horn [3] showed that the same problem can be formulated as a weighted linear least squares problem. Here we present a method which uses orthogonal transformations to solve the Duncan and Horn formulation by taking advantage of the special structure of the problem. This approach gives advantages in numerical accuracy over other related methods in the literature, and is similar in the number of computations required. It also gives a straightforward presentation of the material for those unfamiliar with the area.

1. Introduction. Kalman filtering [9] is a tool used by engineers and others to estimate the behavior of certain linear dynamic systems when both the system, and the measurements made on the system, are subject to zero mean random noise of known covariance, and the initial state of the system comes from a distribution with known mean and covariance. The theory and the algorithms for computing the required estimates were originally developed in an elegant manner using conditional probability theory to obtain the linear estimates that minimized the expected values of the 2-norms of the errors. An excellent introduction to this approach is given by Rhodes [12]. This approach is somewhat involved, and requires some study for a reader not initially familiar with the basic probability theory used. The relation between such minimum variance estimators and weighted least squares solutions led Duncan and Horn [3] to approach the problem in the discrete case from the viewpoint of regression analysis; they showed how the Kalman estimates $x^{(k)}$ could be found by solving a sequence of weighted linear least squares problems

$$\min_{x^{(k)}} \|F^{(k)}x^{(k)}-y^{(k)}\|^2, \qquad k=1,2,\cdots,$$

where $F^{(k)}$ is a known matrix of coefficients, $y^{(k)}$ includes the known measurements, and $\|\cdot\|$ denotes the 2-norm. The problems can be solved in a recursive manner since $F^{(k+1)}$ is just $F^{(k)}$ with additional rows and columns added.

One purpose of this paper is to suggest a method for solving the Duncan and Horn formulation of the problem on a computer. A reliable approach is to compute an orthogonal decomposition of F

$$Q^TF=\begin{bmatrix}R\\0\end{bmatrix}$$

* Received by the editors June 16, 1975, and in revised form April 5, 1976.

† School of Computer Science, McGill University, Montreal, Canada. This work was supported by National Research Council of Canada Grant A8652.

‡ Applied Mathematics Division, Department of Scientific and Industrial Research, Wellington, New Zealand.

where R is upper triangular. When F is not sparse, Q would usually be the product of Householder transformations [2]. This basic approach is used here, but because of the special form of F, the computation is performed more economically using both Householder transformations and Givens rotation matrices (see, for example, [11]). This approach brings out the recursive nature of the solution process very clearly, and leads to straightforward derivations of factors of the inverses of the estimator covariances.

The method, which is described in § 4, has advantages in numerical stability over other published methods for computing the same results. This will be more understandable after the model and method have been presented, so a discussion of some of the methods that are already available, along with a comparison with the present work, will be left until § 9. An operation count is given in § 8.

Another purpose of this paper is to emphasize the importance of Duncan and Horn's large matrix formulation of the linear dynamic estimation problem. This approach shows the essential nature and simplicity of the problem and as the authors say in [3], it "opens the way for further developments in recursive estimation which are more tractable in the regression approach". A derivation of this formulation will be presented in §§ 2 and 3 as a teaching tool.

It is shown in §§ 5 and 6 how the techniques given in § 4 can be used to compute the factors of the inverses of as many covariance matrices as are wanted, and § 7 shows how the method can be adjusted to handle some extensions of the model.

The introduction to the linear dynamic model in § 2 is given in somewhat expanded form to make this paper readable for workers who have some familiarity with techniques of matrix computations, but who are not familiar with this particular problem. Again in the interests of clarity, the computations involving Givens rotations are presented as ordinary rotations. If the size of the problem is such as to justify the overhead involved in using fast, stable, two-multiplication rotations, then descriptions of these can be found in the literature [5], [7], [11].

The notation $E(\cdot)$ will denote *expected value*, and superscript T will denote transpose, otherwise capital italic letters will denote matrices, with the symmetric capitals V, W, etc., reserved for symmetric nonnegative definite matrices. Lower case italic will denote column vectors, except for the indices i, j, k, m, n, p, s, t.

2. Description of the model. Here we will be concerned with dynamic systems, so we will consider a system which is evolving in time. It is assumed that the part of the system we are interested in can be characterized at any given time by a vector of variables, the *state vector*; for example, these variables could be pressure, temperature, etc. The discrete Kalman filter is applicable to linear systems where at time k the system's n dimensional state vector x_k is given, for $k=2,3,\cdots$, by

$$x_k = F'_{k-1}x_{k-1} + B_k u_k + v'_k. \tag{1}$$

Here u_k is a known p dimensional vector which might be chosen to control the system in some desired way, F'_{k-1} and B_k are known matrices, and v'_k is a noise vector which comes from a distribution with zero mean,

$$E(v'_k) = 0, \tag{2}$$

and covariance V_k,

$$E(v_k'v_k'^T) = V_k, \tag{3}$$

where V_k is a known symmetric nonnegative definite matrix.

It is also assumed that x_1, the initial state of the system, is a random vector from a distribution with known mean $x_{1|0}$, the *estimate* of x_1 at time zero, and covariance V_1,

$$E(x_1) = x_{1|0}, \quad E(v_1'v_1'^T) = V_1, \quad v_1' \equiv x_1 - x_{1|0}. \tag{4}$$

Note that this can be written as the equation

$$x_{1|0} = x_1 - v_1' \tag{5}$$

where v_1' is a zero mean noise vector with known covariance V_1.

The problem arises because the state vector x_k is not directly measurable; instead the m dimensional vector of observations y_k' is available where it is known that for $k = 1, 2, \cdots$,

$$y_k' = C_k'x_k + w_k' \tag{6}$$

with C_k' a known matrix and w_k' a noise vector such that

$$E(w_k') = 0, \qquad E(w_k'w_k'^T) = W_k, \tag{7}$$

where W_k is a known nonnegative definite matrix.

Given all the information in (1)–(7) with the observations y_j' available up to time k, we might want to find an estimate of x_k which is best in some sense. This is called *filtering* in the engineering literature (e.g. [12]). We might also want a best estimate of x_{k+j}; for $j > 0$ this is *prediction*, while for $j < 0$ it is *smoothing*.

3. Weighted linear least squares formulation of the problem. Here all the information on the system and measurements up to time k will be given equal weight, and combined together to form one large equation. This can be regarded as one large overdetermined linear system with zero mean unit covariance noise, which can then be solved using linear least squares techniques.

From now on we will assume that the V_j and W_j are all positive definite so that we can transform equations (1), (5) and (6) into equations with zero mean noise vectors having unit covariance matrices, as a result we will be able to assume that all equations are equally important. In (3), (4), and (7) let

$$L_jL_j^T = V_j^{-1}, \qquad \tilde{L}_j\tilde{L}_j^T = W_j^{-1} \tag{8}$$

be the Cholesky decompositions of the inverses of the covariance matrices. If these inverses are available then there is no difficulty. If a covariance matrix V is known, but not its inverse, then the reverse Cholesky decomposition $RR^T = V$, R upper triangular, can be found (see, for example, [11, p. 125]). It then follows that $L \equiv R^{-T}$ is the required Cholesky factor in (8). The reverse decomposition of V has been chosen here as it leads to computational savings later when the method exhibited in (25) is used to triangularize the matrix in (24).

The factor L_1 in (8) can be applied to (5) to give

$$\begin{aligned} d_1 &\equiv L_1^T x_{1|0} = L_1^T x_1 - L_1^T v'_1 \equiv L_1^T x_1 + v_1, \\ E(v_1) &= 0, \qquad E(v_1 v_1^T) = L_1^T V_1 L_1 = I. \end{aligned} \tag{9}$$

Similarly if we write

$$F_{k-1} \equiv -L_k^T F'_{k-1}, \quad d_k \equiv L_k^T B_k u_k, \quad v_k \equiv -L_k^T v'_k \tag{10}$$

then (1) may be rewritten as

$$d_k = [F_{k-1}, L_k^T] \begin{bmatrix} x_{k-1} \\ x_k \end{bmatrix} + v_k, \qquad k = 2, 3, \cdots, \tag{11}$$

with v_k representing a zero mean, unit covariance noise vector.

With the same approach the observation equation (6) becomes

$$y_k \equiv \tilde{L}_k^T y'_k = \tilde{L}_k^T C'_k x_k + \tilde{L}_k^T w'_k \equiv C_k x_k + w_k. \qquad k = 1, 2, \cdots, \tag{12}$$

where w_k is zero mean, unit covariance noise.

Equations (9), (11), and (12) combine to give the formulation of the problem suggested by Duncan and Horn [3]

$$\begin{bmatrix} d_1 \\ y_1 \\ d_2 \\ y_2 \\ d_3 \\ \cdot \\ \cdot \\ y_k \end{bmatrix} = \begin{bmatrix} L_1^T & & & \\ C_1 & & & \\ F_1 & L_2^T & & \\ & C_2 & & \\ & F_2 & \cdot & \\ & & \cdot & \\ & & \cdot & L_k^T \\ & & & C_k \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ x_k \end{bmatrix} + \begin{bmatrix} v_1 \\ w_1 \\ v_2 \\ w_2 \\ v_3 \\ \cdot \\ \cdot \\ w_k \end{bmatrix}. \tag{13}$$

Equation (13) can be written more briefly as

$$y^{(k)} = F^{(k)} x^{(k)} + v^{(k)}, \tag{14}$$

though in the future we will usually drop the superscripts. We see that if for $i \neq j$

$$E(v_i v_j^T) = 0, \quad E(w_i w_j^T) = 0, \quad E(v_i w_j^T) = E(v_i w_i^T) = 0, \tag{15}$$

then $v^{(k)}$ is a zero mean, unit covariance noise vector. Equation (15) will hold if and only if it holds with v and w replaced by v' and w' respectively. From now on we assume (15) holds; this is the assumption originally made by Kalman [9], but its purpose and use is perhaps more easily understandable in the present formulation. Clearly the present approach could be modified to handle correlated noise.

The least squares solution to (14) is

$$\hat{x} \equiv \arg\min_x \|Fx - y\|^2, \tag{16}$$

that is, $\hat{x}$ is the argument that minimizes the sum of squared residuals.

Rosenbrock [13] has observed that Kalman filtering is an extension of the work of Gauss. By putting the problem in the form of (13) and (14), Duncan and Horn have shown that it has *exactly* the same form as the problem that Gauss posed and solved.

4. Numerical solution of the least squares problem. In order to find the solution to (16) we use an orthogonal matrix Q such that

$$Q^TF=\begin{bmatrix}Q_1^T\\Q_2^T\end{bmatrix}F=\begin{bmatrix}R\\0\end{bmatrix} \tag{17}$$

where R is upper triangular. Since the 2-norm is unaffected by orthogonal transformations,

$$\hat{x}=\arg\min\left\|\begin{bmatrix}R\\0\end{bmatrix}x-\begin{bmatrix}Q_1^Ty\\Q_2^Ty\end{bmatrix}\right\|^2, \tag{18}$$

from which it follows that $\hat{x}$ satisfies

$$R\hat{x}=Q_1^Ty. \tag{19}$$

Here R will certainly be nonsingular if the L_j are nonsingular in (13).

This least squares solution for general F was suggested by Golub [6]. Lawson and Hanson [11] treated the case when F has band form. Gentleman [5] showed how fast, numerically stable variants of the Givens rotation matrices without square roots could be used to gain speed in special cases like the present one. Here advantage will be taken of these techniques, and the special form of F, to obtain a fast recursive set of operations for its reduction to block upper bidiagonal form. It will be shown that (19) becomes

$$\begin{bmatrix}R_1 & R_{1,2} & & & \\ & R_2 & R_{2,3} & & \\ & & \cdot & \cdot & \\ & & & R_{k-1} & R_{k-1,k}\\ & & & & \bar{R}_k\end{bmatrix}\begin{bmatrix}x_{1|k}\\x_{2|k}\\\cdot\\x_{k-1|k}\\x_{k|k}\end{bmatrix}=\begin{bmatrix}b_1\\b_2\\\cdot\\b_{k-1}\\\bar{b}_k\end{bmatrix} \tag{20}$$

where $x_{j|k}$ means the weighted linear least squares estimate of x_j, the state vector at time j, given $y_1',\cdots,y_k'$, the observations up to time k. Thus $x_{j|k}$, $j=1,\cdots,k-1$ are smoothed values, $x_{k|k}$ is a filtered value, while $x_{k+1|k}$ would be a predicted value. The filtered value, and as many smoothed values as are needed, can be found directly from (20) by back substitution.

The last block of the matrix R in (20) is $\bar{R}_k$ rather than R_k, since it is the only one of these blocks that will be altered when the decomposition is updated to account for the introduction of new rows and columns corresponding to new observations like (6), or new system equations like (1).

The matrix F in (13) and (14) can be transformed to the matrix R in (17) and (20) in a sequence of $2k-1$ steps. First we set

$$\hat{R}_1\equiv L_1^T; \tag{21}$$

then in steps $2j-1$ and $2j$ we develop an orthogonal matrix Q_j which transforms

$$\begin{bmatrix} \hat{R}_j & 0 \\ C_j & 0 \\ F_j & L_{j+1}^T \end{bmatrix} \quad \text{to} \quad \begin{bmatrix} R_j & R_{j,j+1} \\ 0 & 0 \\ 0 & \hat{R}_{j+1} \end{bmatrix} \tag{22}$$

so that when the zero rows are permuted to the bottom of the right hand matrix, it becomes upper triangular.

In step $2j-1$ an orthogonal matrix $\bar{Q}_j$ is chosen to zero out C_j as follows:

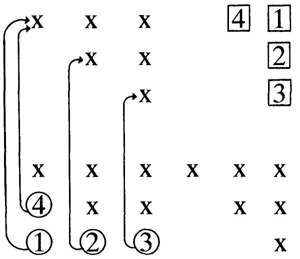
$$\bar{Q}_j^T \begin{bmatrix} \hat{R}_j \\ C_j \end{bmatrix} = \begin{bmatrix} \bar{R}_j \\ 0 \end{bmatrix} \tag{23}$$

where $\bar{R}_j$ is upper triangular and the orthogonal matrix $\bar{Q}_j$ is the product of n Householder matrices chosen to take advantage of $\hat{R}_j$ being upper triangular. Thus for $i = 1, 2, \cdots, n$, the ith elementary matrix will combine row i of $\hat{R}_j$ with the altered C_j matrix to eliminate its ith column. The complete step will take approximately mn^2 multiplications. Equation (20) shows the complete system when a step of this form has just been completed. Next, in step $2j$, the orthogonal matrix $\tilde{Q}_j$ is chosen to zero out F_j:

$$\tilde{Q}_j^T \begin{bmatrix} \bar{R}_j & 0 \\ 0 & 0 \\ F_j & L_{j+1}^T \end{bmatrix} = \begin{bmatrix} R_j & R_{j,j+1} \\ 0 & 0 \\ 0 & \hat{R}_{j+1} \end{bmatrix}, \tag{24}$$

where full advantage of the structure can be taken by using a series of rotations as follows. For $i = n, n-1, \cdots, 1$ the ith row of F_j can be set to zero an element at a time by using the sth row of the altered $\bar{R}_j$ to eliminate the sth element of the ith row of the altered F_j; this must be done in the order $s = 1, 2, \cdots, n$. The start of this step can be indicated schematically in the case of $n = 3$ by

(25)



where ⓢ indicates this element is set to zero in the sth rotation by combining it with the element at the other end of the arrow, and $\boxed{t}$ indicates this originally zero element has been made nonzero in the tth rotation. Other orders of elimination are possible, but this appears to be the quickest, taking about $4n^3$ multiplications using ordinary rotations. If n is large enough to justify the overhead involved in using stable two-multiplication rotations [5], [7], this can be reduced to $2n^3$. More will be said on this in § 8.

The equivalent transformations of the vector y in (13) and (14) will be denoted by setting

$$\hat{b}_1 \equiv d_1, \tag{26}$$

$$\bar{Q}_j^T \begin{bmatrix} \hat{b}_j \\ y_j \end{bmatrix} = \begin{bmatrix} \bar{b}_j \\ r_j \end{bmatrix}, \qquad \tilde{Q}_j^T \begin{bmatrix} \bar{b}_j \\ r_j \\ d_{j+1} \end{bmatrix} = \begin{bmatrix} b_j \\ r_j \\ \hat{b}_{j+1} \end{bmatrix}. \tag{27}$$

Combining these vectors with the matrices in (22), and combining both (23) and (24) gives the computation for the jth time step.

$$Q_j^T \begin{bmatrix} \hat{R}_j & 0 & \hat{b}_j \\ C_j & 0 & y_j \\ F_j & L_{j+1}^T & d_{j+1} \end{bmatrix} = \begin{bmatrix} R_j & R_{j,j+1} & b_j \\ 0 & 0 & r_j \\ 0 & \hat{R}_{j+1} & \hat{b}_{j+1} \end{bmatrix}, \tag{28}$$

$$Q_j^T \equiv \tilde{Q}_j^T \begin{bmatrix} \bar{Q}_j^T & 0 \\ 0 & I \end{bmatrix}, \tag{29}$$

and it is clear that $\hat{R}_{j+1}$ and $\hat{b}_{j+1}$ are ready to be adjoined to C_{j+1} and y_{j+1} etc. for step $2j+1$ and later ones.

At the end of the $2k-1$ steps the vectors r_j in (28) make up the residual of the transformed least squares problem, that is, $Q_2^T y$ in (18). These r_j and the corresponding zero rows of the matrix need only be deleted from the resulting set of equations to produce (20).

We see from (28) that in steps $2j-1$ and $2j$, which correspond to the jth time step, the only information needed from all the previous steps is $\hat{R}_j$ and $\hat{b}_j$. This recursive nature makes the process computationally quite economical. The sparse nature of the matrix in (20) also indicates the economy of storage, since no zero blocks need be stored; in fact if we only want $x_{k-r|k}, \cdots, x_{k|k}$ then R_j, $R_{j,j+1}$, $j=1, \cdots, k-r-1$, can be discarded. Equations (13), (17), and (20) are clear ways of describing the problem and its solution, but it has to be emphasized that such large matrices are not handled during the computation; instead everything is obtained step by step from the start, the important computational transformation being that described by equation (28). Note also that the matrices $\bar{Q}_j$, $\tilde{Q}_j$ and Q_j are never formed in full or stored. With the dimensions of the problem as stated in (1) and (6), step $2j-1$ requires only about $n(n+m)$ storage locations, or less if advantage is taken of the form of $\hat{R}_j$, or if C_j is brought in a row at a time. With careful programming the operations described by (24) could be carried out in about $2n^2$ locations for step $2j$. Thus the only storage that is absolutely necessary during the computation is essentially that required to store the matrices and vectors being dealt with in any particular step.

5. Covariances of the estimators. The estimate $\hat{x}$ of x in (14), which is given by (19), has an error given by

$$\tilde{x} \equiv x - \hat{x} = x - R^{-1}Q_1^T y = x - R^{-1}Q_1^T(Fx+v) = -R^{-1}Q_1^T v \tag{30}$$

which is a random vector with a mean of zero and covariance matrix

$$E(\tilde{x}\tilde{x}^T) = R^{-1}Q_1^T E(vv^T) Q_1 R^{-T} = (R^T R)^{-1} \equiv H. \tag{31}$$

Thus R is the Cholesky factor of H^{-1}. In particular we will be interested in $H_{k|k}$, the covariance of $x_{k|k}$. This is just the bottom right hand corner block of H, which from (31) and the form of R in (20) is just $\bar{R}_k^{-1}\bar{R}_k^{-T}$; thus

$$H_{k|k}^{-1}=\bar{R}_k^T\bar{R}_k \tag{32}$$

so that $\bar{R}_k$ is the Cholesky factor of $H_{k|k}^{-1}$.

Once we have obtained estimates in (13) up to time k via (20), we can add on the next system equation (11)

$$d_{k+1}=F_kx_k+L_{k+1}^Tx_{k+1}+v_{k+1}.$$

This can be combined with (20) in step $2k$, as shown in (24), and the new system solved again. However as this has just added n nonsingular equations in the n unknowns x_{k+1}, it is the equivalent of using (1) to predict

$$x_{k+1|k}=F_k'x_{k|k}+B_{k+1}u_{k+1}, \tag{33}$$

and the previous $x_{j|k}$ will be unchanged. The new matrix corresponding to (20) will have $\hat{R}_{k+1}$ in its bottom right hand corner, and by the same argument as given for $H_{k|k}$ and $\bar{R}_k$, we see that $\hat{R}_{k+1}$ is just the Cholesky factor of $H_{k+1|k}^{-1}$, the inverse of the covariance of the predicted vector $x_{k+1|k}$. That is

$$H_{k+1|k}\equiv E[(x_{k+1}-x_{k+1|k})(x_{k+1}-x_{k+1|k})^T]=(\hat{R}_{k+1}^T\hat{R}_{k+1})^{-1}. \tag{34}$$

To obtain any of the remaining covariance matrices we could reduce R to block lower bidiagonal form using orthogonal transformations from the left, in a sequence of $k-1$ steps commencing at the bottom right hand corner of R. Thus after $k-i$ steps

$$\begin{aligned}
&P_i^TP_{i+1}^T\cdots P_{k-1}^T\left[\begin{array}{cccccc}
R_1 & \cdot & & & & \\
 & \cdot & R_{i-1,i} & & & \\
 & & R_i & R_{i,i+1} & & \\
 & & & R_{i+1} & \cdot & \\
 & & & & \cdot & R_{k-1,k}\\
 & & & & & \bar{R}_k
\end{array}\right]\\
(35)\qquad\\
&=\left[\begin{array}{cccccc}
R_1 & \cdot & & & & \\
 & \cdot & R_{i-1,i} & & & \\
 & & \tilde{R}_i & & & \\
 & & \overset{\approx}{R}_{i+1,i} & \overset{\approx}{R}_{i+1} & & \\
 & & & \cdot & & \\
 & & & & \overset{\approx}{R}_{k,k-1} & \overset{\approx}{R}_k
\end{array}\right]\equiv R^{(i)}
\end{aligned}$$

where $^{\approx}$ indicates this block is in its final form, $\tilde{R}_i$ means this block will be altered in eliminating the block above it, and the remainder have not yet been touched.

From (31) $H_{i|k}$ is the ith $n\times n$ block on the diagonal of

$$(R^TR)^{-1}=[R^{(i)T}R^{(i)}]^{-1}=R^{(i)-1}R^{(i)-T} \tag{36}$$

and we see from (35) that rows $(i-1)n+1$ to in of $R^{(i)-1}$ are

$$[0,\cdots,0,\tilde{R}_i^{-1},0,\cdots,0]$$

and as a result

$$H_{i|k}=\tilde{R}_i^{-1}\tilde{R}_i^{-T}=(\tilde{R}_i^T\tilde{R}_i)^{-1}. \tag{37}$$

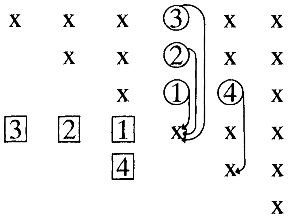
We see then that factors $\tilde{R}_j$ of the inverses of as many error covariance matrices as are wanted may be computed simply by carrying out the bidiagonalization in (35) as far as we want, to produce in sequence $\tilde{R}_k\equiv\bar{R}_k$, and $\tilde{R}_j$, $j=k-1$, $k-2,\cdots$.

Probably the best way of computing each step of this block bidiagonalization of R is to use the reverse of the method (25) that was used for computing (24); in this way full account is taken of sparsity, and the resulting $\tilde{R}_j$ are upper triangular in (37), so these will be the Cholesky factors of the covariance matrix inverses. The next step in (35) can be described as

$$P_{i-1}^T\equiv\begin{bmatrix} I_{n(i-2)} & & \\ & \bar{P}_{i-1}^T & \\ & & I_{n(k-i)} \end{bmatrix},\qquad \bar{P}_{i-1}^T\begin{bmatrix} R_{i-1} & R_{i-1,i} \\ 0 & \tilde{R}_i \end{bmatrix}=\begin{bmatrix} \tilde{R}_{i-1} & 0 \\ \tilde{\tilde{R}}_{i,i-1} & \tilde{\tilde{R}}_i \end{bmatrix} \tag{38}$$

where, for $t=1,2,\cdots,n$, the tth column of $R_{i-1,i}$ is set to zero by a series of rotations of the tth row of $\tilde{R}_i$ with the sth row of the altered $R_{i-1,i}$, $s=n,n-1,\cdots,1$. The start of this can be described as

(39)



where the same notation is used as was used in (25). Again using two-multiplication rotations this takes $2n^3$ multiplications.

This method is consistent with the present approach of computing factors of inverses of covariance matrices directly, and its accuracy is an attractive feature. However it is not clear how the results could be updated from step to step, and so if several of these covariance matrices are wanted regularly, this would be expensive. Another approach is to define the $kn\times n$ matrix N_i having the unit matrix in rows $(i-1)n+1$ to in, and zero elsewhere. Then

$$H_{i|k}=N_i^T(R^TR)^{-1}N_i\equiv S_i^TS_i=R_{i|k}^TR_{i|k}$$

where $R^TS_i=N_i$, and the Cholesky factor $R_{i|k}$ of $H_{i|k}$ could be found by using orthogonal rotations to triangularize S_i. A little thought shows such factors can easily be updated when new information arrives. This appears to be a fast and workable method, the only possible inaccuracies arising in the solution of equations for S_i, if these are ill-conditioned. However this method gives us factors of the covariance matrices, in contrast to the rest of the paper which deals with factors of inverses of covariance matrices. For reasons of space and consistency we will not pursue it further.

6. Further predicted values and their covariances. If any further values of the state vector need be predicted on the basis of the measurements up to and including time k, then these follow directly from (1)

$$x_{k+j|k} = F'_{k+j-1}x_{k+j-1|k} + B_{k+j}u_{k+j}, \tag{40}$$

for $j = 2, 3, \cdots$. These can be thought of as the best estimates for the model

$$\left[\begin{array}{c} y^{(k)} \\ \hdashline d_{k+1} \\ \\ d_{k+j} \end{array}\right] = \left[\begin{array}{cc:ccc} F^{(k)} & & & & \\ \hdashline O & F_k & L_{k+1}^T & & \\ & & \cdot & \cdot & \\ & & & F_{k+j-1} & L_{k+j}^T \end{array}\right] \left[\begin{array}{c} x^{(k)} \\ \hdashline x_{k+1} \\ \cdot \\ x_{k+j} \end{array}\right] + \left[\begin{array}{c} v^{(k)} \\ \hdashline v_{k+1} \\ \cdot \\ v_{k+j} \end{array}\right], \tag{41}$$

which is the continuation of (14) if no further measurements are available. Here the zero mean random noise vector has unit covariance, and so the upper triangular Cholesky factor of the inverse of the covariance matrix of the error $\tilde{x}_{k+j} = x_{k+j} - x_{k+j|k}$ is just the bottom right hand corner matrix obtained when the upper triangularization of $F^{(k)}$ in (17) is continued to the new matrix in (41). This can be done using j steps of the same form as was used for (24).

7. An extension of the model. Although in equations (1) and (6) the state at time $k+1$ and the observation at time k are given in terms of the state at time k only, the model can be more general than this, as is shown for example by Hannan [8]. One possibility is that x_{k+1} can depend linearly on $x_k, x_{k-1}, \cdots, x_{k-s}$ and y_k can depend linearly on $x_k, x_{k-1}, \cdots, x_{k-t}$. The present formulation and computational solution can easily be extended to such cases; for example with $s = t = 1$, the equivalent of (13) for $k = 3$ might be

$$\begin{bmatrix} d_1 \\ y_1 \\ d_2 \\ y_2 \\ d_3 \\ y_3 \end{bmatrix} = \begin{bmatrix} L_1^T & & \\ C_{1,1} & & \\ F_{1,1} & L_2^T & \\ C_{2,1} & C_{2,2} & \\ F_{2,1} & F_{2,2} & L_3^T \\ & C_{3,2} & C_{3,3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} v_1 \\ w_1 \\ v_2 \\ w_2 \\ v_3 \\ w_3 \end{bmatrix}.$$

Here the first three blocks of rows would be transformed to upper triangular form just as in (22); then this complete upper triangle would be used to eliminate $C_{2,1}$ and $C_{2,2}$ as in (23), and the resulting upper triangle used to eliminate $F_{2,1}$ and $F_{2,2}$ and transform L_3^T as in (24). The lower part of the resulting upper triangle would then be used to eliminate $C_{3,2}$ and $C_{3,3}$, and so on.

8. Operation count. A simple generalization of the transformations described by (24), (25) and (38), (39) can be used very effectively in other computations in estimation, as well as in optimal control calculations. The generalization of (24) can be simply described as

$$Q^T\begin{pmatrix} R_1 & 0 \\ G & R_2 \end{pmatrix} = \begin{pmatrix} \bar{R}_1 & \bar{R}_{12} \\ 0 & \bar{R}_2 \end{pmatrix} \tag{42}$$

where Q is orthogonal and R_1 and $\bar{R}_1$ are $n \times n$ upper triangular matrices, but now R_2 and $\bar{R}_2$ are $m \times m$ upper triangular. It is worthwhile considering this slightly more general case in order to compare the use of rotations, as described in § 4, with reduction by Householder transformations.

Using Householder transformations fills up the block R_2 in the first step, and requires $mn(2m+n)$ multiplications to produce upper triangular $\bar{R}_1$, leaving $\bar{R}_2$ full. It then takes another $2m^3/3$ multiplications to make $\bar{R}_2$ upper triangular, a total of

$$mn(2m+n)+2m^3/3 \tag{43}$$

multiplications for the complete Householder reduction. Standard rotations require 4 multiplications each, while it is possible to compute stable two-or three-multiplication rotations [5], [7] with some added overhead. If we use k-multiplication rotations, then the complete reduction can be carried out in

$$kmn(m+n)/2 \tag{44}$$

multiplications. These results show that four-multiplication rotations are faster than Householder transformations when $m > 1.23n$, but in any case never take more than twice as long. In the present case with $m = n$ the comparison is $11n^3/3$ against $4n^3$, so Householder transformations have a (negligible) 8% advantage. An advantage of the technique described in § 4 is that it requires no extra storage, since only one nonzero element is generated as each zero is produced. In contrast, Householder transformations immediately introduce $m^2/2$ nonzero elements that later have to be made zero. For this reason we recommend using rotations at all times: ordinary rotations are satisfactory if there is no time problem, while stable two multiplication rotations can be used if the dimensions of the problem are large enough to warrant them, and these are superior to Householder transformations for any ratio of m and n.

For the complete operation count here, we assume that two multiplication rotations have been used for (24), taking $2n^3$ multiplications. Now if we assume V_j^{-1} and W_j^{-1} are supplied in (8), then computing the Cholesky decompositions and forming $L_j^T F'_{j-1}$ in (10) and $\tilde{L}_j^T C'_j$ in (12) takes $n^3/6+m^3/6+n^3/2+nm^2/2$ multiplications. Computing (23) takes mn^2 multiplications, to give a total for one time step of

$$8n^3/3+m^3/6+mn^2+nm^2/2. \tag{45}$$

This provides the Cholesky factors of the inverses of $H_{j|j}$ and $H_{j+1|j}$. The time required to compute the right hand side in (20) and solve for the estimates $x_{j|k}$ is small in comparison to the other computations.

9. Comparison with other algorithms. In order to compare algorithms it will be useful to discuss briefly some of the important developments since Kalman's original approach. A more complete discussion is given in [10]. The major computation in Kalman's method was to update the covariance matrix $H_{k|k-1}$ in order to produce $H_{k|k}$ and $H_{k+1|k}$ as follows,

$$\begin{aligned} H_{k|k} &= H_{k|k-1} - H_{k|k-1}C_k'^T[C_k'H_{k|k-1}C_k'^T + W_k]^{-1}C_k'H_{k|k-1}, \\ H_{k+1|k} &= F_k'H_{k|k}F_k'^T + V_k, \end{aligned} \tag{46}$$

where we have used the notation in (1), (3), (6), and (7). Matrices which are represented by symmetric capitals here are nonnegative definite in theory, but in computations this property was often lost. Later workers avoided loss of definiteness by updating *square roots* of matrices (where S is said to be a square root of the nonnegative definite matrix H if $H = SS^T$). This also avoided inaccuracies caused by squaring, since it was possible to compute and use CS, the square root of CHC^T, rather than CHC^T itself. This approach led to algorithms described as *covariance square root filters.* If covariance matrices are given in (3) and (7) for the noise, and in (4) for the initial estimate, and if covariance matrices of the estimates are desired as output, then covariance square root filtering appears to be the correct approach.

The inverse of a covariance matrix is called an *information matrix.* It is possible to update information matrices for this problem, but the same arguments on definiteness and accuracy of square roots apply here too, and so it is numerically superior to consider *information square root filters.* The present method requires only factors of information matrices in (8), and gives factors of information matrices in § 5, so it is clearly an information square root filter.

One of the best available approaches for this estimation problem was given by Dyer and McReynolds [4], and this has also been called an information square root filter. They considered zero mean unit covariance noise as in (11) and (12), and sought to minimize

$$J(k) = \|L_1^T(x_1 - x_{1|0})\|^2 + \sum_{j=2}^{k} \|v_j\|^2 + \sum_{j=1}^{k} \|w_j\|^2 \tag{47}$$

subject to the constraints (11) and (12), in order to estimate x_k. An implementation which is as numerically stable as this particular approach allows has been described by Hanson and Dyer [14], and is in use as a navigational tool at the Jet Propulsion Laboratory in Pasadena, California.

Although (47) is a least squares formulation of the problem, Dyer and McReynolds apparently did not consider it in the form of the large sparse matrix problem in (13), and perhaps for this reason they required the orthogonal transformation [4, (26)]

$$P^T\begin{bmatrix} I & 0 \\ \bar{R}_k(F_k')^{-1}L_{k+1}^{-T} & \bar{R}_k(F_k')^{-1} \end{bmatrix} = \begin{bmatrix} N & B \\ 0 & \hat{R}_{k+1} \end{bmatrix} \tag{48}$$

where we have used the terminology of (1), (3), (8), and (32) to (34). The equivalent transformation here is given by (24). The difficulty is that (48) requires the equivalent of the computation of the inverse of F_k', and so cannot be carried out when F'_k is singular, and leads to numerical inaccuracies in general unless F'_k is well-conditioned. Problems with poorly conditioned F'_k are not at all rare. Such difficulties do not occur in the present algorithm using the computation in (24), nor in well designed covariance square root filters, and so this appears to be a serious drawback of the Dyer and McReynolds information square root filter and its implementation in [14].

In one sense the Dyer and McReynolds method is partway between a covariance square root filter and an information square root filter such as the one here, for although Dyer and McReynolds produce factors of information matrices, and require the information matrix of the measurement noise in (6), they require the covariance matrix of the system noise in (1). This suggests the possibility of a complete range of filters extending from the all covariance square root filter to the all information square root filter as outlined here.

An operation count of the Dyer and McReynolds method as described in [4] shows that it takes about

$$m^3/6+m^2n/2+mn^2+16n^3/3 \tag{49}$$

multiplications for the problem described here if the covariance matrix of the system noise and the information matrix of the measurement noise are given. This is $8n^3/3$ more multiplications than in (45), which if $m=n$ means it takes about 60% more multiplications than the method given here. There are other variants of the Dyer and McReynolds approach described in [1] and [10] which give computation counts more comparable with (45) but they also suffer from requiring the inverse of F'_k.

10. Comments. If some observations are noise free the corresponding covariance matrices will be singular. In such cases the method given here will not work (although it can easily be extended to allow for covariance matrices that are singular). Similarly Kalman's original method and other covariance updating methods have difficulties with singular or ill-conditioned information matrices. Thus the different basic methods have complementary applications.

The fact that information filters, such as the one given here, can work in the absence of certain information, that is with singular information matrices, is an important advantage for some cases. Lack of information can cause leading submatrices of R in (20) to be singular; this can be detected in the present algorithm, and in such cases no estimates would be computed. But as more information comes in, nonsingularity of the later submatrices would allow estimates to be computed.

Again we emphasize that the way of formulating the problem suggested by Duncan and Horn is a natural and easily understandable approach and leads directly to good computational techniques. The well known fact that this weighted least squares result gives the best linear unbiased estimate for the case of nonsingular, bounded, noise covariance matrices, ties the result to Kalman's work without having to show the equivalence algebraically. The speed of the algorithm given here, together with its good numerical properties, makes it an attractive one when compared with other published algorithms.

Acknowledgments. We are very grateful for the support of the Department of Scientific and Industrial Research, Wellington where the initial part of this work was done, and for the hospitality of the people working there, especially Garry Dickinson whose interest lead to our reading the paper by Duncan and Horn. We also thank Roger Govindarajan for programming the algorithm, and Mike Arbib, Gerald Bierman, Richard Hanson, and George Styan whose comments helped our understanding of the subject.

REFERENCES

[1] G. J. BIERMAN, *Sequential square root filtering and smoothing of discrete linear systems*, Automatica—J. IFAC, 10 (1974), pp. 147–158.

[2] P. BUSINGER AND G. H. GOLUB, *Linear least squares solutions by Householder transformations*, Numer. Math., 7 (1965), pp. 269–276.

[3] D. B. DUNCAN AND S. D. HORN, *Linear dynamic recursive estimation from the viewpoint of regression analysis*, J. Amer. Statist. Assoc., 67 (1972), pp. 815–821.

[4] P. DYER AND S. MCREYNOLDS, *Extension of square-root filtering to include process noise*, J. Optimization Theory Appl., 3 (1969), pp. 444–458.

[5] W. M. GENTLEMAN, *Least squares computations by Givens transformations without square roots*, J. Inst. Math. Appl., 12 (1973), pp. 329–336.

[6] G. GOLUB, *Numerical methods for solving linear least squares problems*, Numer. Math., 7 (1965), pp. 206–216.

[7] S. HAMMARLING, *A note on modifications to the Givens plane rotations*, J. Inst. Math. Appl., 13 (1974), pp. 215–218.

[8] E. J. HANNAN, *Multiple Time Series*, John Wiley, New York, 1970.

[9] R. E. KALMAN, *A new approach to linear filtering and prediction problems*, Trans. ASME Ser. D J. Basic Engrg., 82D (1960), pp. 35–45.

[10] P. G. KAMINSKI, A. E. BRYSON AND S. F. SCHMIDT, *Discrete square root filtering: a survey of current techniques*, IEEE Trans. Automatic Control, AC–16 (1971), pp. 727–736.

[11] C. L. LAWSON AND R. J. HANSON, *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, N.J., 1974.

[12] I. B. RHODES, *A tutorial introduction to estimation and filtering*, IEEE Trans. Automatic Control, AC–16 (1971), pp. 688–706.

[13] H. H. ROSENBROCK, *Sur les relations entre les filtres linéaires discrets et quelques formules de Gauss*, Identification, Optimalisation et Stabilité des Systèmes Automatiques, Actes du Congrès d'Automatique théorique, Paris, 1965, pp. 239–249.

[14] R. J. HANSON AND P. DYER, *A computational algorithm for sequential estimation*, Comput. J., 14 (1971), pp. 285–290.