Information Formulation of the UDU Kalman Filter

Christopher D’Souza and Renato Zanetti

Abstract

A new information formulation of the Kalman filter is presented where the information matrix is parameterized as the product of an upper triangular matrix, a diagonal matrix, and the transpose of the triangular matrix (UDU factorization). The UDU factorization of the Kalman filter is known for its numerical stability, this work extends the technique to the information filter. A distinct characteristic of the new algorithm is that measurements can be processed as vectors, while the classic UDU factorization requires scalar measurement processing, i.e. a diagonal measurement noise covariance matrix.

I. INTRODUCTION

The UDU formulation of the Kalman Filter has been used in aerospace engineering applications for several decades. Thornton [1], Bierman and Thornton [2] and Bierman [3] introduced an elegant formulation where the covariance matrix $P$ is replaced by two factors: a diagonal matrix $D$ and an upper triangular matrix $U$ with ones on the main diagonal, such that $P = UDU^T$. Whereas the UDU factorization improves the computational stability and efficiency of large navigation filters, it was originally used in a batch formulation [4]. However, this formulation lent itself to sequential implementations, well-suited for platforms where both computational stability and numerical efficiency are at a premium. It serves as the backbone of the Orion Navigation System [5].

Factorization of the covariance matrix in a Kalman filter [6] is almost as old as the filter itself. In 1963 James Potter developed a square-root formulation of the Kalman filter to implement on the Apollo onboard computer [7]. The main driver at the time was numerical precision, as computer words were only 8 bits long. Replacing the covariance by a square root matrix $S$, such as $P = SS^T$, reduces the spread of the elements of $P$ bringing them closer to 1, doubling the numerical precision of the stored variable. Potter’s algorithm requires the computation of scalar square roots (one per measurement). At the time, the Apollo Kalman filter was designed without any process noise, because computations required for inclusion of the process noise required too many computations [8]. A very desirable by-product of this factorization is that the symmetry and semi-positive definiteness of the covariance are insured by construction, and does not need to be checked or enforced to correct for numerical and round-off errors. It should be noted that this Apollo factorization was not a triangular square root matrix.

An alternative square root covariance factorization is the Cholesky factorization [9], [10]. The Cholesky method is very similar to Potter’s but computes the square root of the covariance matrix with a Cholesky decomposition ($S$ is a triangular matrix) [11]. Another relevant covariance factorization work is that proposed by Oshman and Bar-Itzhack [12], which utilizes the spectral decomposition of the covariance matrix.

The UDU factorization is not a square root filter; the numerical precision of the stored variable does not increase due to the factorization. For example, if $P$ is diagonal, $U = I$ and $D = P$; therefore the full range of values in $P$ are preserved in this factorization. However, the $UDU$ formulation of the Kalman filter has great numerical stability properties [3]; it insures symmetry of the covariance by construction, and it requires a trivial check and correction to ensure semi-positive definiteness (it suffices to enforce that the diagonal elements of $D$ remain non-negative). The UDU formulation is free from square root operations, making it computationally cheaper than the Cholesky approach. For these reasons the UDU has endured as one of the preferred practical implementation of Kalman filters in aerospace applications.

While the UDU factorization is well known, it has never been applied to the information formulation of the Kalman filter [13], [14], [15]. In this formulation the inverse of the covariance matrix, known as the information matrix, is carried in the recursive algorithm rather than the covariance matrix itself. The information formulation is a popular approach in several situations. In particular, the Square Root Information Filter (SRIF) [16], [17], [18], [3], [19], [20] is a go-to Kalman filter factorization method used in orbit determination packages such as Monte because of its great stability and accuracy. In this work we introduce the UDU Information Filter, a never developed before algorithm with two key properties that make it a very desirable implementation of a recursive estimator: i. unlike the regular UDU filter, measurements do not need to be processed as scalars, i.e. the measurement noise covariance matrix $R$ does not need to be diagonal or diagonalized, and ii. unlike the regular information formulation the state estimation error covariance matrix does not actually need to be inverted.

C. D’Souza is with the Aerospace and Flight Mechanics Division, EG6, 2101 NASA Parkway, NASA Johnson Space Center, Houston, Texas 77058.
R. Zanetti is with the Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, Texas 78712.
II. BACKGROUND

The well-known Kalman filter measurement update equations are given by

\[
\begin{align}
\bar{x}_k &= \bar{x}_k + K_k(y_k - H_k \bar{x}_k) \tag{1} \\
P_k &= \bar{P}_k - \bar{P}_k H_k^T (H_k \bar{P}_k H_k^T + R_k)^{-1} H_k \bar{P}_k \tag{2} \\
K_k &= \bar{P}_k H_k^T (H_k \bar{P}_k H_k^T + R_k)^{-1} = \bar{P}_k H_k^T R_k^{-1} \tag{3}
\end{align}
\]

where the bar represents the a priori value, \(K_k\) is the \(n \times m\) Kalman gain, \(x \in \mathbb{R}^n\) is the state vector, \(P_k\) is the \(n \times n\) estimation error covariance matrix, \(y \in \mathbb{R}^m\) is the measurement vector defined as

\[
y_k = H_k x_k + \eta_k \tag{4}
\]

where \(\eta_k\) is a zero mean, white sequence with covariance matrix \(R_k\). The propagation equations are

\[
\begin{align}
\bar{x}_{k+1} &= \Phi(t_{k+1}, t_k) \bar{x}_k \tag{5} \\
\bar{P}_{k+1} &= \Phi(t_{k+1}, t_k) \bar{P}_k \Phi(t_{k+1}, t_k)^T + G_k Q_k G_k^T \tag{6}
\end{align}
\]

where \(\Phi(t_{k+1}, t_k)\) (which we will denote as \(\Phi_k\)) is the state \(n \times n\) transition matrix from \(t_k\) to \(t_{k+1}\), \(Q_k\) is the \(p \times p\) process noise covariance matrix, and \(G_k\) is the \(n \times p\) process noise shaping matrix.

The UDU factorization implements the above equations by replacing the covariance matrix \(P_k\) with an upper triangular matrix with ones on the diagonal (\(U_k\)) and a diagonal matrix \(D_k\), such that

\[
P_k = U_k D_k U_k^T \tag{7}
\]

The UDU approach to propagate \(U_k\) and \(D_k\) forward in time makes use of the Modified Weighted Gram-Schmidt (MWGS) orthogonalization algorithm that avoids loss of orthogonality due to round-off errors [21]. Measurements are processed one at the time as scalars by noting that when \(R_k\) is diagonal the update in Eq. (2) is obtained by recursively processing one element of \(y_k\) at a time, using the corresponding row of \(H_k\) and diagonal element of \(R_k\). The measurement residual covariance matrix \(W_k = H_k \bar{P}_k H_k^T + R_k\) thus becomes a scalar, and the quantity \(\bar{P}_k H_k^T = \bar{w}_k\) becomes a vector; thus each of the scalar updates takes the form

\[
P_k = \bar{P}_k - \frac{1}{\bar{w}_k} \bar{w}_k \bar{w}_k^T \tag{8}
\]

since matrix \(\bar{P}_k\) is updated with a rank one matrix \((\frac{1}{\bar{w}_k} \bar{w}_k \bar{w}_k^T)\), we call this a rank one update, Agee and Turner [22] detailed how to directly update the \(U_k\) and \(D_k\) factors due to a rank one update. The subtraction in Eq. (8) could cause some numerical instabilities in Agee-Turner’s algorithm. Carlson [23] introduced an alternative rank-one update algorithm that, while less generic, is more stable for the measurement update. Carlson’s rank-one update is not valid for generic values of \(\bar{w}_k\) and \(W_k\), but only when the update is done with the optimal Kalman gain.

An alternative formulation of the Kalman filter is the information formulation, where the covariance matrix \(P\) is replaced by its inverse. The covariance update and Kalman gain are calculated as [13]

\[
\begin{align}
P_k^{-1} &= \bar{P}_k^{-1} + H_k^T H_k R_k^{-1} \tag{9} \\
K_k &= \bar{P}_k H_k^T R_k^{-1} \tag{10}
\end{align}
\]

The information formulation is particularly useful when there is no prior information, i.e. \(P_0 = \infty\), in this case the covariance formulation of the KF is not defined, while the information formulation is, and starts from \(P_0^{-1} = 0\). In the covariance formulation, the \(m \times m\) measurement residual covariance matrix \(W_k = H_k \bar{P}_k H_k^T + R_k\) is inverted to process the measurement, while in the information formulation the \(n \times n\) covariance matrix is inverted in the time propagation step. In situations when \(m > n\), therefore, the information formulation could be computationally cheaper, although measurements are often processed one at the time as scalars. Processing measurements as scalars is only possible when \(R_k\) is diagonal, otherwise the additional steps of a change of variables to diagonalize \(R_k\) is required.

III. THE UDU INFORMATION FILTER

A. The Measurement Update

Begin with factorizing the covariance \(P\) into an LDL form; that is, rather than using an upper triangular matrix we will use a lower triangular matrix. We denote the diagonal matrix with \(\Delta\)

\[
P_k = L_k \Delta_k L_k^T \quad \text{and} \quad \bar{P}_k = \bar{L}_k \bar{\Delta}_k \bar{L}_k^T \tag{11}
\]
and we define $U$ and $D$ as the inverses of $L^T$ and $\Delta$, respectively

$$P_k^{-1} = L_k^{-T} \Delta_k^{-1} L_k^{-1} = U_k D_k U_k^T$$  
(12)  
$$P_k = L_k^{-T} \Delta_k^{-1} L_k^{-1} = U_k D_k U_k^T$$  
(13)

so that the measurement update (Eq. (9)) becomes

$$U_k D_k U_k^T = U_k D_k U_k^T + H_k^T R_k^{-1} H_k$$  
(14)

We now factorize the $m \times m$ matrix $R_k$ into LDL form as

$$R_k = L_{R_k} \Delta_{R_k} L_{R_k}^T$$  
(15)

and

$$R_k^{-1} = L_{R_k}^{-T} \Delta_{R_k}^{-1} L_{R_k}^{-1} = U_{R_k} D_{R_k} U_{R_k}^T$$  
(16)

so that Eq. (14) becomes

$$U_k D_k U_k^T = U_k D_k U_k^T + H_k^T U_{R_k} D_{R_k} U_{R_k}^T H_k$$  
(17)

We now work on the term $U_{R_k} H$ where we note that it is of dimension $m \times n$ so that it can be expressed as

$$U_{R_k}^T H = \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_m^T \end{bmatrix}$$  
(18)

where each $v_i$ is an $n \times 1$ vector.

The factor $H_k^T R_k^{-1} H_k$ can be expressed as

$$H_k^T R_k^{-1} H_k = H_k^T U_{R_k} D_{R_k} U_{R_k}^T H_k$$

$$= \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_m^T \end{bmatrix} \begin{bmatrix} 1/d_{kR} & 0 & \cdots & 0 \\ 0 & 1/d_{2R} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1/d_{mR} \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_m^T \end{bmatrix}$$

$$= \sum_{i=1}^{m} \frac{1}{d_{iR}} v_i v_i^T$$  
(19)

so that the measurement update equation is now

$$U_k D_k U_k^T = U_k D_k U_k^T + \sum_{i=1}^{m} \frac{1}{d_{iR}} v_i v_i^T$$  
(20)

Thus it reduces to a series of $m$ rank-one updates.

Notice that Eq. (20) is the update equation due to a vector measurement so that the need to process scalar measurements, as in the covariance UDU formulation, is avoided in the proposed information UDU formulation. Rather than performing an eigenvalue decomposition of $R_k$ and a corresponding change of variables for $y_k$, we simply perform the UDU factorization of $R_k^{-1}$.

As stated earlier, one of the benefits of using an information formulation is that if $P_0$ is singular, this allows for an estimate to be obtained [14]. When $P_0$ is singular, $x_0$ is not completely defined. To this end, $\tilde{x}_k$ and $\bar{x}_k$, which are directly related to $\hat{x}_k$ and $\bar{x}_k$, are defined as

$$\tilde{x}_k \overset{\triangle}{=} P_k^{-1} \bar{x}_k, \quad \bar{x}_k \overset{\triangle}{=} \bar{P}_k^{-1} \tilde{x}_k$$  
(21)

Premultiplying Eq. (1) by $P_k^{-1}$, we get

$$\dot{\tilde{x}}_k = P_k^{-1} (I - K_k H_k) \tilde{x}_k + P_k^{-1} K_k y_k$$  
(22)

$$\tilde{x}_k + H_k^T R_k^{-1} y_k$$  
(23)
B. The Time Update

Prior to propagation, the standard information formulation of the Kalman filter inverts the information matrix to obtain the covariance, it then propagates the covariance with Eq. (6), and finally inverted the propagated covariance matrix to prepare for the measurement update phase. We propose an algorithm that propagates the factors of the information matrix directly. Starting from the covariance propagation Eq. (6), we factorize $Q_{k}$ via a UDU parameterization so that

$$Q_{k} = U_{Q_k} \Delta Q_{k} U_{Q_k}^T$$

(24)

where $\Delta Q_{k}$ is a diagonal $p \times p$ matrix and define $G_{Q_k}$ as

$$G_{Q_k} \triangleq G_{k} U_{Q_k}$$

(25)

so that $\overline{P}_k$ becomes

$$\overline{P}_{k+1} = \Phi_{k} P_{k} \Phi_{k}^T + G_{Q_k} \Delta Q_{k} G_{Q_k}^T$$

(26)

Invoking the matrix inversion lemma

$$(Z + XAY)^{-1} = Z^{-1} - Z^{-1} X (A^{-1} + Y Z^{-1} X)^{-1} Y Z^{-1}$$

(27)

and letting

$$Z = \Phi_{k} P_{k} \Phi_{k}^T; \quad A = \Delta Q_{k}; \quad X = G_{Q_k}; \quad Y = G_{Q_k}^T$$

(28)

and

$$Z^{-1} = M_{k} \triangleq \Phi_{k}^{-T} P_{k}^{-1} \Phi_{k}^{-1}$$

(29)

The inverse of the propagated covariance is

$$\overline{P}_{k+1}^{-1} = M_{k} - M_{k} G_{Q_k} \left[ G_{Q_k}^T M_{k} G_{Q_k} + \Delta_{Q_k}^{-1} \right]^{-1} G_{Q_k}^T M_{k}$$

(30)

Defining

$$\overline{G}_{k} \triangleq \Phi_{k}^{-1} G_{Q_k} \quad D_{Q_k} = \Delta_{Q_k}^{-1}$$

(31)

$P_{k+1}^{-1}$ becomes

$$\overline{P}_{k+1}^{-1} = \Phi_{k}^{-T} \left\{ P_{k}^{-1} - \right.$$

$$P_{k}^{-1} \overline{G}_{k} \left[ \overline{G}_{k}^T P_{k}^{-1} \overline{G}_{k} + D_{Q_k} \right]^{-1} \overline{G}_{k}^T P_{k}^{-1} \left. \right\} \Phi_{k}^{-1}$$

(32)

and defining $K_k$ as

$$K_k \triangleq P_{k}^{-1} \overline{G}_{k} \left[ \overline{G}_{k}^T P_{k}^{-1} \overline{G}_{k} + D_{Q_k} \right]^{-1}$$

(33)

Defining the quantity inside the brackets in Eq. (32) as $\overline{P}_{k}^{-1}$

$$\overline{P}_{k}^{-1} \triangleq P_{k}^{-1} - P_{k}^{-1} \overline{G}_{k} \left[ \overline{G}_{k}^T P_{k}^{-1} \overline{G}_{k} + D_{Q_k} \right]^{-1} \overline{G}_{k}^T P_{k}^{-1}$$

(34)

We notice Eq. (34) is an analog to Eq. (2) with

$$P_{k}^{-1} \rightarrow P_{k} \quad \overline{G}_{k} \rightarrow H_{k}^T \quad D_{Q_k} \rightarrow R_{k}$$

$$P_{k}^{-1} \rightarrow \overline{P}_{k} \quad K_{k} \rightarrow K_{k}$$

and since $D_{Q_k}$ is a diagonal $p \times p$ matrix, we can solve for the UDU factorization of $\overline{P}_{k}^{-1}$ directly by using a Carlson Rank-One Update [23] performed $p$ times on

$$U_{k} D_{k} U_{k}^T = U_{k} D_{k} U_{k}^T -$$

$$U_{k} D_{k} U_{k}^T \overline{G}_{k} \left[ \overline{G}_{k}^T U_{k} D_{k} U_{k}^T \overline{G}_{k} + \Delta_{Q_k}^{-1} \right]^{-1} \overline{G}_{k}^T U_{k} D_{k} U_{k}$$

(35)

so that we can find the time-propagated UDU factors of $\overline{P}_{k}^{-1}$ as

$$\overline{P}_{k+1}^{-1} = \overline{Q}_{k+1} \overline{D}_{k+1} \overline{U}_{k+1}^T \quad = \quad \Phi_{k}^{-T} U_{k} D_{k} U_{k}^T \Phi_{k}^{-1}$$

(36)
Since this equation is equivalent to a covariance propagation without process noise, the MWGS orthogonalization algorithm can be used to obtain the factors $U_{k+1}$ and $D_{k+1}$.

Notice that $\Phi_k^{-1}$ does not necessarily need to be computed by a direct matrix inversion. Usually $\Phi_k$ is computed via integration of a matrix differential equation or by series approximation. Similarly, $\Phi_k^{-1}$ can be obtained directly by backwards integration or by series approximation.

Beginning with Eq. (5) the time update for $z_{k+1}$ is obtained as follows

$$P_{k+1}^{-1} \bar{x}_{k+1} = \bar{P}_{k+1} \Phi_k P_k P_k^{-1} \bar{x}_k$$

which becomes

$$\bar{z}_{k+1} = \bar{P}_{k+1}^{-1} \Phi_k P_k \bar{z}_k$$

or

$$\bar{z}_{k+1} = \Phi^{-T}(t_{k+1}, t_k) \left[ I - K_k \bar{G}_k^T \right] \bar{z}_k$$

The following table summarizes the UDU Information Filter. While Table I contains a compact notation for the covariance time propagation and measurement update; in the actual algorithm the covariance factors $U_t$ and $D_t$ are individually propagated and updated using the Rank-1 Update and the Modified Weighted Gram-Schmidt orthogonalization algorithms.

<table>
<thead>
<tr>
<th>Initialization</th>
<th>State</th>
<th>$z_0 = P^{-1}(t_0) \bar{x}(t_0)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Covariance</td>
<td>$U_0 D_0 U_0^T = P^{-1}(t_0)$</td>
</tr>
<tr>
<td>Truth</td>
<td>$x_{k+1} = \Phi_k x_k + G_k \nu_k$, $\nu_k \sim n(0, Q_k)$</td>
<td></td>
</tr>
<tr>
<td>Process Noise</td>
<td>$U_{k+1} D_{k+1} U_{k+1}^T = Q_k$, $G_k = \Phi_k^T G_k U_k Q_k$</td>
<td></td>
</tr>
<tr>
<td>Gain</td>
<td>$K_k = U_k D_k U_k^T G_k G_k^T U_k D_k U_k^T G_k + \Delta Q_k$</td>
<td></td>
</tr>
<tr>
<td>Covariance</td>
<td>$U_{k+1} D_{k+1} U_{k+1}^T = I - K_k \bar{G}_k^T \left[ I - \bar{K}_k \bar{G}_k^T \right] \bar{z}_k$</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>$x_{k+1} = \Phi_{k+1}^{-1} \left[ I - \bar{K}_k \bar{G}_k^T \right] \bar{z}_k$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Measurement Update</th>
<th>Truth</th>
<th>$y_{k+1} = H_{k+1} x_k + \eta_{k+1}$, $\eta_{k+1} \sim n(0, R_{k+1})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meas. Noise</td>
<td>$U_{R_{k+1}+1} D_{R_{k+1}+1} U_{R_{k+1}+1}^T R_{k+1} = R_{k+1}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\sum_{i=1}^{m} \nu_i \nu_i^T = H_{k+1}^T U_{R_{k+1}+1} D_{R_{k+1}+1} U_{R_{k+1}+1}^T H_{k+1} + \sum_{i=1}^{m} (1/d_{i,k}) \nu_i \nu_i^T$ (Rank-1 Update)</td>
<td></td>
</tr>
<tr>
<td>Covariance</td>
<td>$U_{k+1} D_{k+1} U_{k+1}^T = U_{k+1} D_{k+1} U_{k+1}^T k_{k+1} + \sum_{i=1}^{m} (1/d_{i,k}) \nu_i \nu_i^T$ (Rank-1 Update)</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>$\bar{z}<em>{k+1} = \bar{z}</em>{k+1} + H_{k+1} \bar{R}<em>{k+1} \bar{y}</em>{k+1}$</td>
<td></td>
</tr>
</tbody>
</table>

**Table I**

**SUMMARY OF UDU INFORMATION FILTER EQUATIONS**

C. An Efficient Algorithm to compute $U^{-1}$

The algorithm proposed does not necessitate to invert the covariance matrix nor its $U$ or $L$ factor. However, in case the initial covariance was provided, it might be convenient to factorize it first, and to efficiently invert its factors rather than inverting the full covariance. In this section we compute the inverse in an efficient manner, taking advantage of the ‘1’s’ and ‘0’s’. It is as follows: Given an $n \times n$ upper triangular ‘unit’ matrix $U$ expressed as

$$U = \begin{bmatrix}
1 & U_{1,2} & U_{1,3} & \cdots & U_{1,n-1} & U_{1,n} \\
0 & 1 & U_{2,3} & \cdots & U_{2,n-1} & U_{2,n} \\
0 & 0 & 1 & \cdots & U_{3,n-1} & U_{3,n} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 1 & U_{n-1,n} \\
0 & 0 & 0 & \cdots & 0 & 1
\end{bmatrix}$$

(41)
the inverse is also an \( n \times n \) upper triangular ‘unit’ matrix \( V \) (so that \( \det(V) = 1 \)) which is

\[
V = U^{-1} = \begin{bmatrix}
1 & V_{1,2} & V_{1,3} & \cdots & V_{1,n-1} & V_{1,n} \\
0 & 1 & V_{2,3} & \cdots & V_{2,n-1} & V_{2,n} \\
0 & 0 & 1 & \cdots & V_{3,n-1} & V_{3,n} \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 1 & V_{n-1,n} \\
0 & 0 & 0 & \cdots & 0 & 1
\end{bmatrix}
\]  

(42)

since

\[
UV = I
\]  

(43)

and the \( ij \)-th element of \( UV \) is given by

\[
\sum_{k=1}^{j} U_{i,k}V_{k,j} = U_{i,i}V_{i,j} + U_{i,j}V_{j,j} + \sum_{k=i+1}^{j-1} U_{i,k}V_{k,j}
\]  

(44)

we can solve for the elements of \( V \) as

\[
V_{i,j} = - \left[ U_{i,j} + \sum_{k=i+1}^{j-1} U_{i,k}V_{k,j} \right]
\]  

(45)

IV. A Numerical Example

In this section we show the performance of the algorithm in linear, time-varying example with correlated measurement noise covariance matrix \( R \). The system is given by

\[
x_{k+1} = \Phi_k x_k + \nu_k
\]

(46)

\[
y_k = H_k x_k + \eta_k
\]

(47)

\[
\Phi_k = \begin{bmatrix}
I & A_k \\
B_k & I
\end{bmatrix}
\]

(48)

\[
A_k = \begin{bmatrix}
t_k - t_{k-1} & 0 \\
0 & t_k - t_{k-1}
\end{bmatrix}
\]

(49)

\[
B_k = 0.1 \begin{bmatrix}
\sin(t_k) - \sin(t_{k-1}) & -(\cos(t_k) - \cos(t_{k-1})) \\
0 & \sin(t_k) - \sin(t_{k-1})
\end{bmatrix}
\]

(50)

\[
H_k = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0
\end{bmatrix}
\]

(51)

where \( I \) is the identity matrix \( t_k - t_{k-1} = 1 \) second, \( \nu_k \) is a zero mean, Gaussian white sequence with covariance matrix \( Q_k = 0.01 I \), and \( \eta_k \) is a zero mean, Gaussian white sequence with covariance matrix \( R_k \)

\[
R_k = \begin{bmatrix}
2.96 & 2.8 \\
2.8 & 2.96
\end{bmatrix}
\]

(52)

The initial estimate is unbiased, and the initial estimation error is Gaussian with covariance \( P_0 = I \). Fig. 1 shows the result of a single run and the \( 3\sigma \) predicted standard deviations from a straight formulation of a Kalman filter. In order to show the equivalence between the Kalman filter (KF) and the UDU information approach (UDUI), Fig. 2 shows the norm of the difference between the two state estimates

\[
\epsilon_x = \| \hat{x}_{KF} - \hat{x}_{UDUI} \|
\]

while Fig. 3 compares the Kalman filter covariance \( P_{KF} \) with the UDU factorization of the Information matrix \( P_{UDUI}^{-1} = UDU^T \) by plotting the following quantity:

\[
\epsilon_P = \| P_{KF} P_{UDUI}^{-1} - I \|
\]

The figures show that the proposed algorithm results closely match the Kalman filter hence validating the proposed algorithm as its UDU information formulation. The growth of the error in Figs. 2 and 3 is due to the accumulation of round-off errors in the algorithms. It is known [3] that numerical errors accumulate faster in the full covariance formulation of the Kalman filter than in the UDU’s.
Fig. 1. Estimation Error and $3\sigma$ predicted standard deviations

Fig. 2. Norm of the State Error ($\|\hat{x}_{KF} - \hat{x}_{UDUI}\|$)

Fig. 3. Norm of the Covariance error ($\|P_{KF} P_{UDUI}^{-1} \! - \! I\|$)

V. CONCLUSIONS

A new algorithmic mechanization of the classic Kalman filter is presented, the new algorithm combines the information formulation with the UDU factorization. While the covariance formulation of the Kalman filter is usually employed, the information formulation has distinct advantages in some applications, for example when no initial condition is available. The UDU factorization is a widely adopted technique to produce a numerically stable and accurate algorithm to keep the covariance matrix symmetric and positive definite. A numerical example confirms the equivalency between the Kalman filter and the proposed algorithm.

ACKNOWLEDGMENTS

This work was supported in part by NASA JSC contract NNX17AI35A.
REFERENCES


