Square-Root Information Filter and Smoother

Rosemary Huang

October 28, 2010

Abstract

An essential part of the operation of any spacecraft is orbit determination (OD), calculation of its state. This is done to ensure that the spacecraft is moving the way that it is expected to, and if not corrections can be made. However, it is difficult to do this accurately because there are uncertainties that come from errors in the measurements, the dynamic model, and the initial estimate. Several algorithms are available for OD that take into account these uncertainties to make an estimate of a spacecraft's state, and one of these is square root information filter (SRIF). This algorithm has been incorporated into Code 595's Orbit Determination Toolbox (ODTBX), a software package that is intended to be used in preliminary mission design. As a mission progresses, more and more measurements are made so that the estimate of its state becomes more accurate. These additional measurements also make it possible to refine past state estimates. Many algorithms are available for this as well, one of these is a modification of the SRIF called the square root information smoother (SRIS). A SRIS has also been implemented in ODTBX.

1 Introduction

Orbit determination encompasses the whole process of measuring and calculating a spacecraft's state, which includes its position and velocity, in order to find what kind of orbit it is in. This process is also known as estimation, because one is never completely certain of the actual state. There is uncertainty in the state that results from modeling errors, measurement noise, and uncertainty in the initial estimate.

Filters and smoothers are algorithms that give the best estimate of the state inspite of these uncertainties. Figure **??** gives a general illustration of the way these algorithms operate. A filter works forward in time by taking in measurements and providing a solution that is a least squares fit, one that minimizes the sum of the squares of the errors. This solution includes an estimate of the state estimate and a covariance matrix that indicates the confidence in this estimate. As time progresses, a filter takes in more and more measurements and more and more confidence is gained in the estimate, which hopefully it is closer to the unknown true spacecraft state. If a better estimate of a state in the past is desired, a smoother can be used. A smoother is essentially a backwards filter. It works by taking the current state estimate and covariance, which has all the measurements already incorporated into it, and propagates it backwards in time to the point of interest.

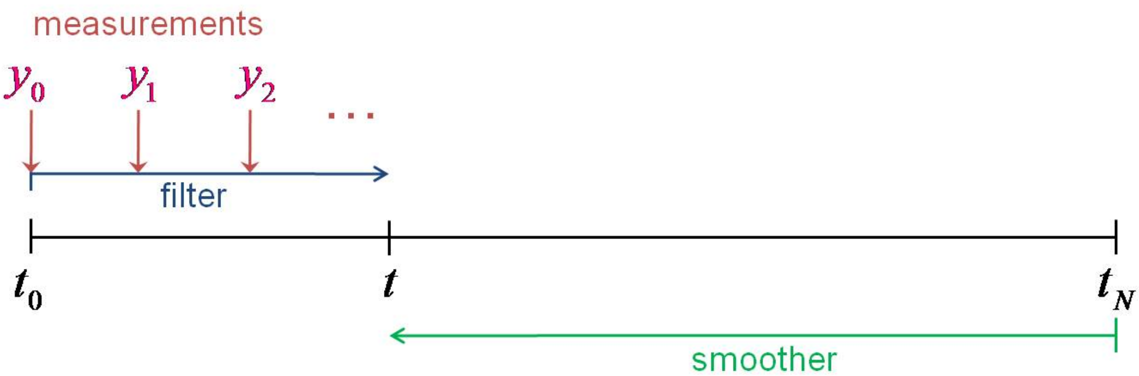

Figure 1: Schematic of Filtering and Smoothing.

There are may types of filter/smoother algorithms. The most well known types of algorithms, such as Kalman Filter, are based on updating the estimate covariance. Others are based on updating the information matrix, which is the inverse of the covariance matrix. Both these types of algorithms run into stability problems due to round-off error that is a consequence of the limited numerical precision of computers. Square-root algorithms are a way to mitigate this problem. By taking the square root of the either the covariance or information matrix, the precision of storage is doubled, and numerical stability is improved.

This paper describes a filter/smoother algorithm based on the square root of the information matrix, which is known conveniently as a square-root information filter/smoother (SRIF/S). It was implemented in Code 595's Orbit Determination Toolbox, a software package for MATLAB intended to be used in preliminary mission design analysis.

2 Algorithm Description

In this section, the square-root information filter/smoother algorithms are described beginning first with the definitions of the model, estimate equation, measurement equation, and process noise estimate equations, which are then converted to a specific form for use in the algorithms. Then, the steps of the square-root information filter are explained. Finally, the smoother is described.

2.1 Definitions

2.1.1 Dynamic Model with Process Noise

The SRIF/S algorithms use the following discrete dynamic model.

$$\begin{gathered} x_k = \Phi(t_k, t_j)x_j + \Gamma(t_k, t_j)u_j \\ E[u_j] = \bar{u} \;\; E[(u_j - \bar{u})(u_k - \bar{u})^\top] = Q\delta_{j,k} \end{gathered} \tag{1}$$

where x_k and x_j represent the state vector with n number of states at two different times t_k and t_j. The state transition matrix $\Phi(t_k, t_j)$ maps x_j to time t_k. Also present is $\Gamma(t_k, t_j)$, which maps the process noise u_j vector of length p to time t_k. The mean of the process noise is $\bar{u}$, while the covariance of u is a product of the process noise covariance matrix Q and the Kroniker delta $\delta_{j,k}$. Here, Q is assumed to be positive definite.

2.1.2 State Estimate and the Data Equation

The state estimate can be represented in the following form:

$$\begin{gathered}\bar{x} = x + \eta \\ E[\eta] = 0 \quad E[\eta\eta] = P = \Lambda^{-1}\end{gathered} \tag{2}$$

where $\bar{x}$ is the mean of the estimate of x, and the error in $\bar{x}$ is η, which is zero mean with a covariance of P. Note that the inverse of P is the information matrix Λ. This representation has to be changed into another form for use in the SRIF/S. This is done by first finding a root R of the information matrix such that $\Lambda = R^\top R$, which relates to the error covariance matrix as $P = R^{-1}R^{-\top}$. Then the multiplying the state estimate, Equation **??**, by R results in

$$\begin{gathered}z = Rx + \bar{\eta} \\ [\bar{\eta}] = 0 \quad E[\bar{\eta}\bar{\eta}^\top] = I\end{gathered} \tag{3}$$

Here, $z := R\bar{x}$ and $\bar{\eta} := R\eta$. The estimate in this form is known as a data equation, and the set $\{z, R\}$ is called an information pair.

2.1.3 Measurement Equations

The measurements can be expressed by the following equation:

$$\begin{aligned} y &= Hx + \nu \\ E[\nu] = 0 \ \ E[\nu\nu^\top] &= P_\nu \end{aligned} \tag{4}$$

where y is the vector of measurements, H is the measurement partials matrix, and ν is the measurement noise vector. The measurement noise is zero mean and has a covariance of P_ν. Like the state estimate, this also has to be converted to a special form by whitening the measurements. This is done by taking a root L_ν of the covariance that satisfies $P_\nu = L_\nu L_\nu^\top$ and using it throughout the measurement equation (**??**) to get the following

$$\begin{aligned} y_w &= H_w x + \nu_w \\ E[\nu_w] = 0 \ \ E[\nu_w\nu_w] &= I \end{aligned} \tag{5}$$

with $y_w := L_\nu^{-1}y$, $H_w := L_\nu^{-1}H$, and $\nu_w := L_\nu^{-1}\nu$.

2.1.4 Process Noise Estimate and Data Equation

A representation of the process noise estimate $\bar{u}$ is also necessary. It is as follows:

$$\begin{aligned} \bar{u} &= u + \alpha \\ E[\alpha] = 0 \ \ E[\alpha\alpha^\top] &= Q \end{aligned} \tag{6}$$

This must also be converted into a form similar to that in Equation **??** by taking the root R_u of the inverse of Q such that $Q = R_u^{-1}R_u^{-\top}$. Then with $z_u := R_u\bar{u}$ and

$\bar{\alpha} := R_u \alpha$ the process noise data equation is

$$\begin{aligned} z_u &= R_u x + \bar{\alpha}_u \\ E[\bar{\alpha}_u] = 0 \;\; & E[\bar{\alpha}_u \bar{\alpha}_u] = I \end{aligned} \tag{7}$$

2.2 Filtering Steps

The square-root information filter has two main steps. First is a data processing step where new measurements are incorporated into the estimate. Second is a mapping step where a prediction of the estimate at the next time step is calculated. These steps are repeated until the desired end time.

2.2.1 Data Processing

In this step the *a priori* information pair $\{\bar{z}_k, \bar{R}_k\}$ that was propagated to t_k from the last time step t_{k-1} and combined with measurement at the current time step t_k to solve for an updated estimate. Similar to other filter algorithms, the solution is found through the minimization of a least squares performance functional. In this case, the performance functional is set up to minimize the sum of the squares of the estimation error η_k and the measurement error $\nu_{w,k}$.

$$J_k(x) = ||\eta_k||^2 + ||\nu_{w,k}||^2 = ||\bar{R}_k x_k - \bar{z}_k||^2 + ||\bar{H}_{w,k} x_k - \bar{y}_{w,k}||^2 \tag{8}$$

Through some manipulation, this cost can also be expressed in terms of an augmented system consisting of the data equation and whitened measurement equation

$$J_k(x) = \left|\left| \begin{bmatrix} \bar{R}_k \\ \bar{H}_{w,k} \end{bmatrix} x_k - \begin{bmatrix} \bar{z}_k \\ \bar{y}_{w,k} \end{bmatrix} \right|\right|^2 \tag{9}$$

Any orthogonal transformation on the system will not change the cost $J(x)$ since it is within a norm. By performing an orthogonal transformation so that the first n columns are triangularized the updated estimate is found.

$$J_k(x) = \left\| T \begin{bmatrix} \bar{R}_k \\ \bar{H}_{w,k} \end{bmatrix} x_k - T \begin{bmatrix} \bar{z}_k \\ \bar{y}_{w,k} \end{bmatrix} \right\|^2 = \left\| \begin{bmatrix} \hat{R}_k \\ 0 \end{bmatrix} x_k - \begin{bmatrix} \hat{z}_k \\ e_k \end{bmatrix} \right\|^2 \tag{10}$$

The top n rows give the updated information pair $\{\hat{z}_k, \hat{R}_k\}$, which has indirectly updated the state estimate. To see this, notice that the square of the norm transformed system can be into two separate norms as follows

$$J_k(x) = ||e_k||^2 + ||\hat{R}_k x_k - \hat{z}_k||^2 \tag{11}$$

Then to minimize $J(x)$ the second norm has to be zero, which gives the state estimate, $\hat{x}_k = \hat{R}_k^{-1} \hat{z}_k$.

The augmented system is represented in the SRIF as an augmented matrix on which the orthogonal transformation is performed

$$\hat{T}_k \begin{bmatrix} \bar{R}_k & \bar{z}_k \\ H_{w,k} & y_{w,k} \end{bmatrix} = \begin{bmatrix} \hat{R}_k & \hat{z}_k \\ 0 & e_k \end{bmatrix} \tag{12}$$

The SRIF implemented in ODTBX uses QR decomposition to perform the triangularizing orthogonal transformation.

2.2.2 Mapping to the Next Time Step

This algorithm step takes the *a posteriori* information pair $\{\hat{z}_k, \hat{R}_k\}$ from the data processing step at time t_k and maps it to the next time step t_{k+1}. How this is done depends on whether process noise is being considered.

No Process Noise To obtain the equations for the mapping step without process noise, recall that for the sequential estimator (Kalman filter) the propagation of the estimate $\hat{x}_k$ and covariance $\hat{P}_k$ to the next time step t_{k+1} is performed through these relations

$$\bar{x}_{k+1} = \Phi(t_{k+1}, t_k)\hat{x}_k$$
$$\bar{P}_{k+1} = \Phi(t_{k+1}, t_k)\hat{P}_k\Phi^\top(t_{k+1}, t_k)$$

Since $P = R^{-1}R^{-\top}$ then

$$\bar{R}_{k+1}^{-1}\bar{R}_{k+1}^{-\top} = \Phi(t_{k+1}, t_k)\hat{R}_k^{-1}\hat{R}_k^{-\top}\Phi^\top(t_{k+1}, t_k)$$

which means

$$\bar{R}_{k+1}^{-1} = \Phi(t_{k+1}, t_k)\hat{R}_k^{-1}$$

and

$$\bar{R}_{k+1} = \hat{R}_k\Phi^{-1}(t_{k+1}, t_k)$$

Additionally, $z = Rx$ for an optimal state estimate, so

$$\begin{aligned}
\bar{z}_{k+1} &= \bar{R}_{k+1}\bar{x}_{k+1} \\
&= \hat{R}_k\Phi^{-1}(t_{k+1}, t_k)\Phi(t_{k+1}, t_k)\hat{x}_k \\
&= \hat{R}_k\hat{x}_k = \hat{z}_k
\end{aligned}$$

Thus, the mapping step for no process noise is simply

$$\begin{aligned} \bar{z}_{k+1} &= \hat{z}_k \\ \bar{R}_{k+1} &= \hat{R}_k \Phi^{-1}(t_{k+1}, t_k) \end{aligned} \tag{13}$$

With Process Noise When process noise is being considered, another approach is required. Similar to the data processing step, the essence of this step involves the minimization of another least squares performance functional. Except in this case, it is a sum of the squared estimation error η_k, measurement error $\nu_{w,k}$, and process noise error $\bar{\alpha}_k$, which are all at time t_k.

$$\begin{aligned} \hat{J}_k(t) &= ||\eta_k||^2 + ||\nu_{w,k}||^2 + ||\bar{\alpha}_k||^2 \\ &= ||\bar{R}_k x_k - \bar{z}_k||^2 + ||H_{w,k} x_k - y_{w,k}||^2 + ||R_u u_k - \bar{z}_{u,k}||^2 \end{aligned} \tag{14}$$

The first two terms have already been optimized through the data processing step already described, which means that they can be replaced by the right hand side of Equation **??**

$$\hat{J}_k(t) = ||e_k||^2 + ||\hat{R}_k x_k - \hat{z}_k||^2 + ||R_u u_k - \bar{z}_{u,k}||^2$$

From the dynamic model, it is known that $x_k = \Phi^{-1}(t_{k+1}, t_k)[x_{k+1} - \Gamma(t_{k+1}, t_k) u_k]$. Also let $\tilde{R}_{k+1} := \bar{R}_k \Phi^{-1}(t_{k+1}, t_k)$. Then the performance functional can be rewritten as

$$\hat{J}_k(t) = ||e_k||^2 + ||\tilde{R}_{k+1} \Phi^{-1}(t_{k+1}, t_k)[x_{k+1} - \Gamma(t_{k+1}, t_k) u_k] - \hat{z}_k||^2 + ||R_u u_k - \bar{z}_{u,k}||^2$$

The last two norms can be expressed as a norm of a single augmented system

$$\hat{J}_k(t) = ||e_k||^2 + \left\| \begin{bmatrix} R_u & 0 \\ -\tilde{R}_{k+1}\Gamma(t_{t+1}, t_k) & \tilde{R}_{k+1} \end{bmatrix} \begin{bmatrix} u_k \\ x_{k+1} \end{bmatrix} - \begin{bmatrix} z_{u,k} \\ \hat{z}_k \end{bmatrix} \right\|^2$$

Now another orthogonal transformation $\bar{T}_{k+1}$ is applied to the augmented system within the norm to triangularize the first p columns to obtain

$$\hat{J}_k(t) = ||e_k||^2 + \left\| \begin{bmatrix} \bar{R}_{u,k+1} & \bar{R}_{ux,k+1} \\ 0 & \bar{R}_{k+1} \end{bmatrix} \begin{bmatrix} u_k \\ x_{k+1} \end{bmatrix} - \begin{bmatrix} \bar{z}_{u,k+1} \\ \bar{z}_{k+1} \end{bmatrix} \right\|^2$$

As with the data processing step, this procedure can be performed on an augmented matrix as follows

$$\bar{T}_{k+1} \begin{bmatrix} R_u & 0 & z_{u,k} \\ -\tilde{R}_{k+1}\Gamma(t_{k+1}, t_k) & \tilde{R}_{k+1} & \hat{z}_k \end{bmatrix} = \begin{bmatrix} \bar{R}_{u,k+1} & \bar{R}_{ux,k+1} & \bar{z}_{u,k+1} \\ 0 & \bar{R}_{k+1} & \bar{z}_{k+1} \end{bmatrix} \qquad (15)$$

The a priori information pair $\{\bar{R}_{k+1}, \bar{z}_{k+1}\}$ for the next time step can be extracted from the last n rows of the transformed matrix to be used in the next data processing step.

2.3 Smoothing Steps

The square-root information smoother is essentially a backwards version of the mapping with process noise step in the SRIF. It relates the smoothed information pair $\{\tilde{z}^*_{j+1}, R^*_{j+1}\}$ from a time step t_{j+1} to the one before it. The noise matrices R_u, R_{ux} and transition matrices Φ, Γ are saved from the filtering steps. Another orthogonal transformation T^*_N is found to triangularize the matrix. This solves the system so that the smoothed information pair $\{\tilde{z}_j^*, R_j^*\}$ at the previous time t_j is in the last n rows.

$$
\begin{aligned}
T_N^* \begin{bmatrix} \bar{R}_{u,j} + \bar{R}_{ux,j}\Gamma(t_{j+1}, t_j) & \bar{R}_{ux,j}\Phi(t_{j+1}, t_j) & \bar{z}_{u,j} \\ R_{j+1}^*\Gamma(t_{j+1}, t_j) & R_{j+1}^*\Phi(t_{j+1}, t_j) & \tilde{z}_{j+1}^* \end{bmatrix} \\
= \begin{bmatrix} R_{u,j}^* & R_{ux,j}^* & z_{u,j}^* \\ 0 & R_j^* & z_j^* \end{bmatrix}
\end{aligned}
\tag{16}
$$

The initial condition for the smoother is the final information pair from the filter, $\tilde{z}_N^* = z_N^* = \hat{z}_N$, and $R_N^* = \hat{R}_N$. This step is then repeated until the desired time point is reached, which is t_0 as implemented in ODTBX.

3 Implementation Results

The square root information filter/smoother was implemented in ODTBX and tested on an internally built test case. Before the smoother can be run, the filter has to run first. The results from the filter are shown in Figure **??**. It shows the errors of the estimate from the true trajectory as a blue line, and the covariance of the estimate as a green line.

Figure **??** shows the results from the smoother run on the same test case after the filter was done. Notice that the covariance of the estimate is smaller that the covariance from the filter. This is consistent with what is expected with a working smoother.

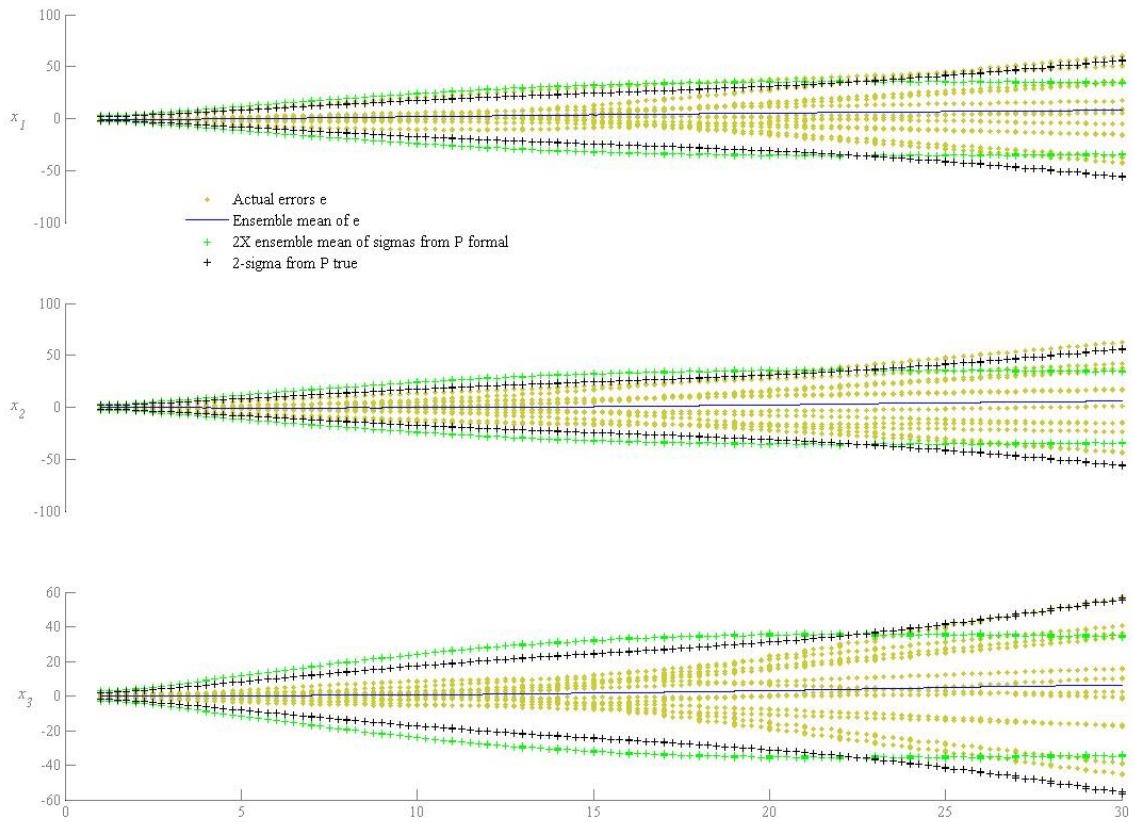

Figure 2: Results from SRIF.

4 Conclusion

A square-root information smoother was added to Code 595's Orbit Determination Toolbox for MATLAB. It was added to code that already existed for the square-root information filter. The smoother was then tested on an internal test case. A comparison of the estimation covariances between the filter and the smoother is consistent with expected behavior, and the smoother is working properly.

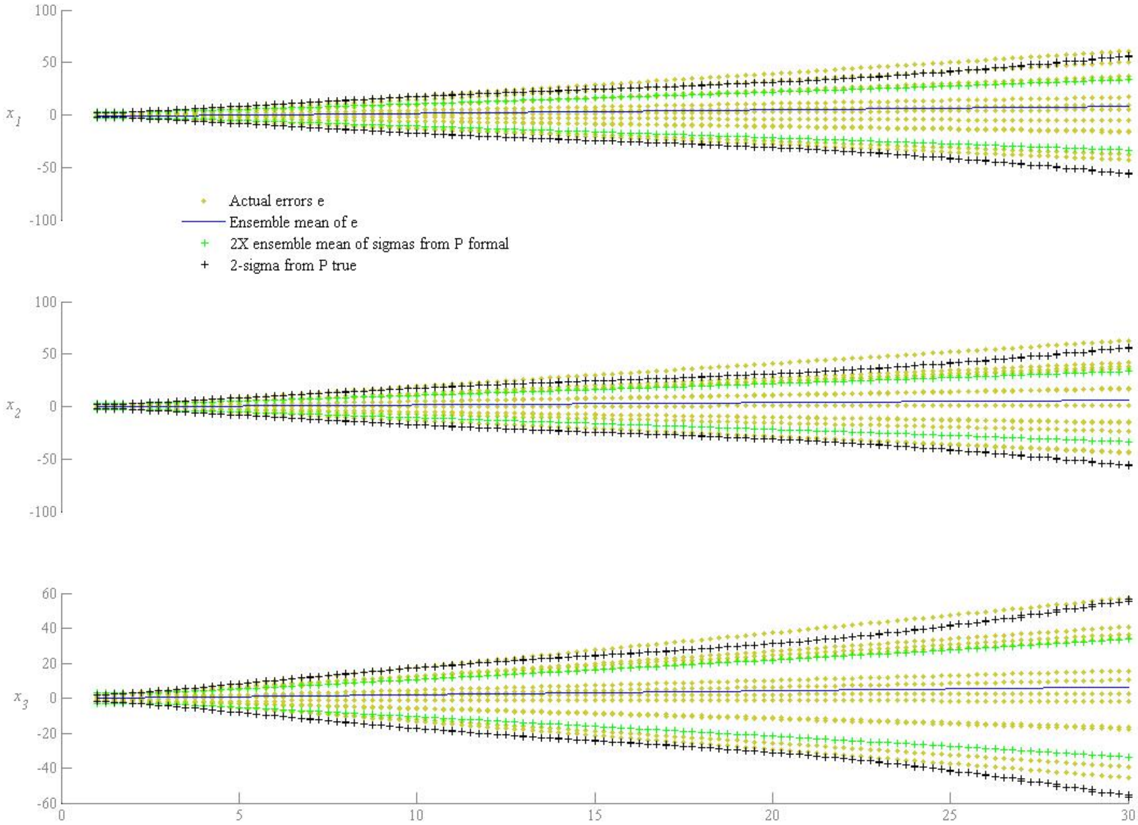

Figure 3: Results from SRIS.

References

[1] Bierman, G.J., *Factorization Methods for Discrete Sequential Estimation*, Academic Press, New York, 1977.

[2] Tapley, B.D., Schutz, B.E., and Born, G.H., *Statistical Orbit Determination*, Elsevier Academic Press, New York, 2004.

[3] Campbell, L.A., Cook, J.W., Cunningham, K.E., and Feess, W.A,"Experiences in Implementation and Use of the Square Root Information Filter for Orbit

Determination," *Proceedings of the 27th Conference on Decision and Control, Austin, Texas*, IEEE, 1988, pp. 831-839.

[4] Gelb, Arthur, *Applied Optimal Estimation*, The M.I.T. Press, Cambridge, Massachusetts, 1989.