Scientific
Research

Sensor Fusion with Square-Root Cubature Information Filtering

Ienkaran Arasaratnam
Ford Motor Company of Canada Ltd., Windsor, ON, Canada
Email: haran@ieee.org

ABSTRACT

This paper derives a square-root information-type filtering algorithm for nonlinear multi-sensor fusion problems using the cubature Kalman filter theory. The resulting filter is called the square-root cubature Information filter (SCIF). The SCIF propagates the square-root information matrices derived from numerically stable matrix operations and is therefore numerically robust. The SCIF is applied to a highly maneuvering target tracking problem in a distributed sensor network with feedback. The SCIF's performance is finally compared with the regular cubature information filter and the traditional extended information filter. The results, presented herein, indicate that the SCIF is the most reliable of all three filters and yields a more accurate estimate than the extended information filter.

Keywords: Cubature Kalman Filter; Information Filter; Multi-Sensor Fusion; Square-Root Filtering

1. Introduction

Sensor fusion is generally defined as the use of techniques that combine data from multiple sensors such that the resulting information is more accurate and more reliable than that from a single sensor. Sensor fusion techniques are widely used in many applications such as mobile robot navigation, surveillance, air-traffic control and intelligent vehicle operations. For multiple sensor fusion in a linear Gaussian environment, the information filter, which can be considered as the dual of the Kalman filter, has been a viable solution [1-3]. Although the Kalman filter and the information filter are algebraically equivalent, the Kalman filter propagates a state vector and its error covariance whereas the information filter uses an information vector and an information matrix. This difference makes the information filter to be superior to the Kalman filter in fusion problems because computations are straight-forward and simple. Moreover, no prior information about the system state is required.

For nonlinear fusion problems however, it is difficult to obtain an optimal solution. In the past, researchers have closely followed the linear fusion theory to obtain a suboptimal solution for nonlinear fusion problems. Recently, Vercauteren *et al*. derived the sigma-point information filter using the statistical linear regression theory and the unscented transformation [4] whereas Kim *et al*. derived a similar set of steps using a minimum mean squared-error criterion [5]. When we are confronted with an issue of striking the appropriate balance or trade-off

between accuracy and computational complexity, the cubature Kalman filter (CKF) is considered to be the logical choice [6]. The CKF is a more accurate and stable estimation algorithm than the unscented/sigma-point filter. However, due to the fact that the sigma-point filter and the CKF share a number common characteristics, the derivation of the Cubature Information Filter (CIF) is straightforward and hence trivial. In this paper, we focus on to derive the square-root version of the CIF for improved numerical stability. Unlike the CIF, the SCIF avoids numerically sensitive matrix operations such as matrix square-rooting and inversion.

The rest of the paper is organized as follows: Section 2 reviews information filtering in general. Section 3 derives the SCIF using the linear fusion theory and matrix algebra. In order to validate the formulation and reliability of the SCIF, it is applied to a highly maneuvering target tracking problem in a distributed sensor network with feedback in Section 4. Section 5 concludes the paper with final remarks.

2. Information Filtering: A Brief Review

The information filter is a modified version of the Kalman filter. The state estimates and their corresponding covariances in the Kalman filter are replaced by the information vectors and information matrices (inverse covariances), respectively, in the information filter. The updated covariance and the updated state take the information form, as shown by

$$Y_{k|k} = P_{k|k}^{-1} \tag{1}$$

$$\hat{y}_{k|k} = P_{k|k}^{-1}\hat{x}_{k|k}. \tag{2}$$

Similarly, the predicted covariance and the predicted state have equivalent information forms:

$$Y_{k+1|k} = P_{k+1|k}^{-1} \tag{3}$$

$$\hat{y}_{k+1|k} = P_{k+1|k}^{-1}\hat{x}_{k+1|k} \tag{4}$$

At the heart of any one of information filters lies the information update, which now becomes a trivial sum.

$$Y_{k+1|k+1} = Y_{k+1|k} + \mathcal{I}_{k+1} \tag{5}$$

$$\hat{y}_{k+1|k+1} = \hat{y}_{k+1|k} + i_{k+1} \tag{6}$$

Here, the information contribution matrix and the information contribution vector are defined as follows, respectively [4,5]:

$$\mathcal{I}_{k+1} = \left(Y_{k+1|k}P_{xz,k+1|k}\right)R_{k+1}^{-1}\left(Y_{k+1|k}P_{xz,k+1|k}\right)^{\mathrm{T}} \tag{7}$$

$$i_{k+1} = \left(Y_{k+1|k}P_{xz,k|k}\right)R_{k+1}^{-1}\left(z_{k+1} - \hat{z}_{k+1|k} + P_{xz,k+1|k}^{\mathrm{T}}\hat{y}_{k+1|k}\right) \tag{8}$$

Information Fusion

In the sensor fusion literature, there are a number of sensor networks with their own virtues and limitations [2]. In this paper, we specifically consider a distributed configuration with feedback [3]. As shown in **Figure 1**, in this network, each local sensor has its own information processor. The locally processed results are then transmitted to the fusion center for computing a global estimate. The global estimate is broadcast so that all the local sensors utilize the global estimate for the purpose of processing the next measurement. The advantage of using the information filters within the local sensors is that the global estimate in the fusion center can be computed from n_s sensor measurements at each time step by simply summing the local information vectors and matrices:

$$\hat{y}_{k|k} = \hat{y}_{k|k-1} + \sum_{s=1}^{n_s} i_{k,s} \tag{9}$$

$$Y_{k|k} = Y_{k|k-1} + \sum_{s=1}^{n_s} \mathcal{I}_{k,s} \tag{10}$$

Note that the computations outlined in this section hold under the following assumptions: 1) The tracking problem at hand is described by a linear Gaussian system; 2) The sensor measurements are uncorrelated to each other; 3) There is no measurement origin of ambiguity; 4) The sensors are synchronized; 5) There is no receipt of out-of sequence measurements; 6) There is no communication loss among sensors. When one or more of these conditions are violated, various techniques have been proposed to get around them in the literature [2].

3. Square-Root Cubature Information Filtering

In each recursion cycle, it is important that we preserve the two properties of an information matrix, namely, its positive definitiveness and symmetry. Unfortunately, when the CIF is committed to an embedded system with limited word-length, numerical errors may lead to a loss of these properties. The accumulation of numerical errors may cause the information filter to diverge or otherwise crash. The CIF involves numerically sensitive operations such as matrix square-rooting and matrix inversion, which may combine to destroy the fundamental properties of an information matrix. The logical procedure to preserve both properties of the information matrix and to improve the numerical stability is to design a square-root version of the CIF. Although the square-root cubature information filter (SCIF) is reformulated to propagate the square-roots of the information matrices, both the CIF and the SCIF are algebraically equivalent to other.
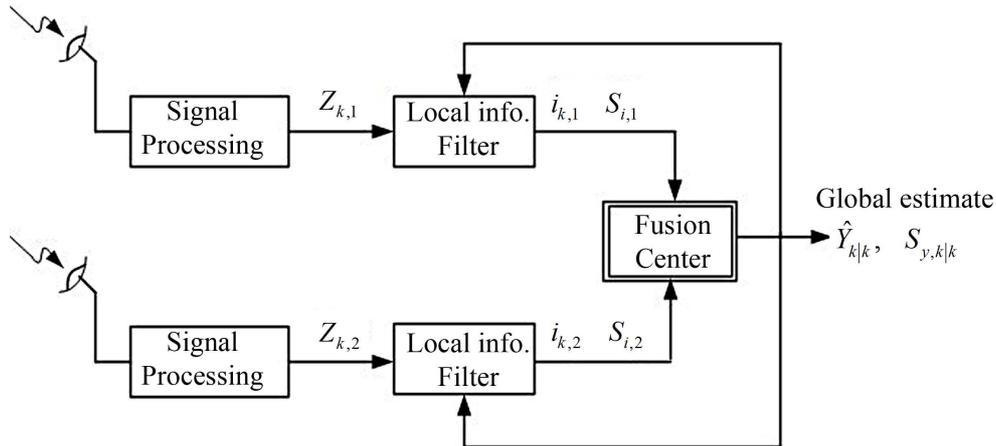
Figure 1. Information flow in a distributed sensor configuration with feedback.

Before deriving the SCIF, for convenience, we introduce the following notations:

- We denote a general triangularization algorithm (e.g., QR decompositio n) as $S = \text{Tria}(A)$, where S is a lower triangular matrix. The matrices A and S are related to each other as follows: Let R be an upper triangular matrix obtained from the QR decomposition on A^T; then $S = R^\mathrm{T}$.
- We use $S_{Q,k}$ and $S_{R,k}$ to denote the square-roots of Q_k and R_k, respectively. That is, $Q_k = S_{Q,k} S_{Q,k}^\mathrm{T}$ and $R_k = S_{R,k} S_{R,k}^\mathrm{T}$.

Because many of the SCIF computations can be easily borrowed from the SCKF [7], we only derive the steps that require explicit treatments in the sequel. Like the SCKF, the SCIF also includes two steps, namely, the time update and the measurement update.

3.1. Time Update

Let $Y_{k|k}$ be

$$Y_{k|k} = S_{y,k|k} S_{y,k|k}^\mathrm{T} \tag{11}$$

and $Y_{k+1|k}$ be

$$Y_{k+1|k} = S_{y,k+1|k} S_{y,k+1|k}^\mathrm{T}. \tag{12}$$

As depicted in **Figure 2**, the time update of the SCIF connects $\left(\hat{y}_{k|k}, S_{y,k|k}\right)$ to $\left(\hat{y}_{k+1|k}, S_{y,k+1|k}\right)$ via three paths. Let us first consider how to derive Path 1, in which the information space is projected onto the state space. Taking inverse on both sides of (1), we get

$$P_{k|k} = Y_{k|k}^{-1}. \tag{13}$$

Substituting the square-root factors on both side of (13) yields

$$S_{k|k} S_{k|k}^\mathrm{T} = \left(S_{y,k|k} S_{y,k|k}^\mathrm{T}\right)^{-1} = S_{y,k|k}^{-\mathrm{T}} S_{y,k|k}^{-1} \tag{14}$$

We may therefore write the square-root of the error covariance matrix

$$S_{k|k} = S_{y,k|k}^{-\mathrm{T}} \tag{15}$$

We summarize the following important result from (13) and (15) as follows:

$$\boxed{P_{k|k} = Y_{k|k}^{-1}} \Rightarrow \boxed{S_{k|k} = S_{y,k|k}^{-\mathrm{T}}}. \tag{16}$$

Because $S_{y,k|k}$ is a triangular matrix, the least-squares method can be used to avoid computing its inversion explicitly [8]. From (2), we write the updated state estimate

$$\hat{x}_{k|k} = P_{k|k} \hat{y}_{k|k} = S_{k|k} S_{k|k}^\mathrm{T} \hat{y}_{k|k}, \tag{17}$$

which completes Path 1.

As shown in **Figure 2**, Path 3 is an inverse projection (from the state space to the information space) of Path 1. By closely following this idea and using (3), (4), the state space quantities can be projected back onto the information space to obtain Path 3. Because Path 2 is identical to the time update of the SCKF, the reader may refer to Section VII of [7] for a detailed derivation.

3.2. Measurement Update

In the measurement update step, we will see how to fuse a new measurement with the predicted information to obtain the updated information. As depicted in **Figure 3**, the measurement update step also includes three paths. Consider Path 1. Given $\hat{x}_{k+1|k}$ and $S_{k+1|k}$, the predicted measurement $\hat{z}_{k+1|k}$ and the sub-matrices of the transformation matrix T, namely, T_{11} and T_{21} can be obtained as described in Section VII of [7]. Fortunately, $\hat{x}_{k+1|k}$ and $S_{k+1|k}$ are available as the by-products of the time update of the SCIF, specifically, from Path 2 of the time update. For this reason, we do not describe the derivation of Path 1 in this paper.
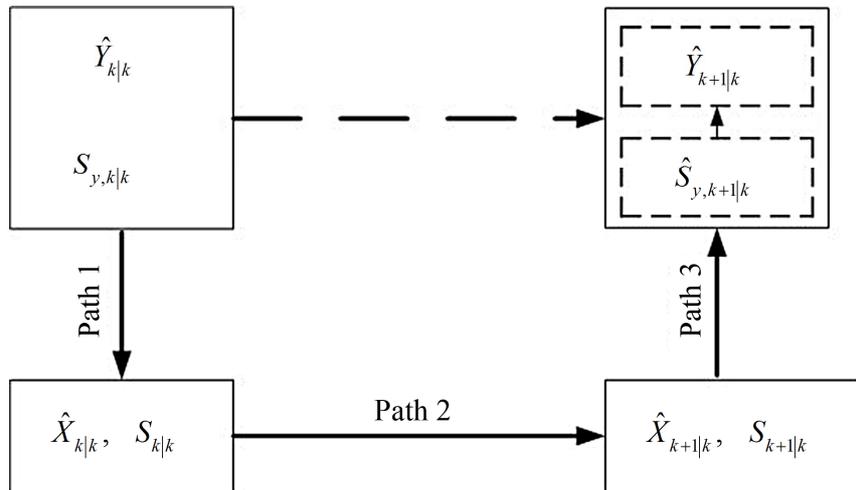
Figure 2. Time update of the SCIF.
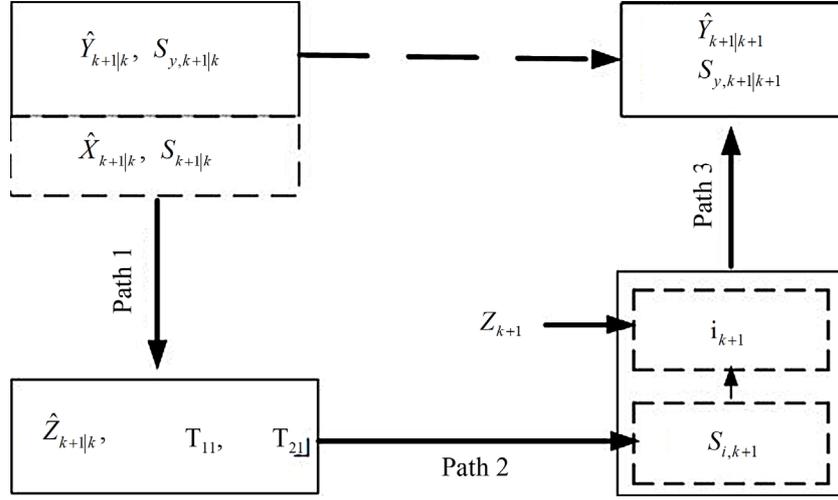
Figure 3. Measurement update of the SCIF.

Let us derive Path 2 now. The end products of Path 2 are the information contribution vector (i_{k+1}), and the square-root information contribution matrix $(S_{i,k+1})$. First, we will derive $S_{i,k+1}$. By closely following (16), we may write the inverse of the measurement noise covariance matrix

$$\overline{S}_{R,k} = S_{R,k}^{-\mathrm{T}}. \qquad (18)$$

Substituting (18) into (7) and rearranging the right hand side, we get

$$\mathcal{I}_{k+1} = \left(Y_{k+1|k} P_{xz,k+1|k} \overline{S}_{R,k+1}\right)\left(Y_{k+1|k} P_{xz,k+1|k} \overline{S}_{R,k+1}\right)^{\mathrm{T}} \qquad (19)$$

Therefore, from (19), we may write $\mathcal{I}_{k+1}$ in a factored form:

$$\mathcal{I}_{k+1} = S_{i,k+1} S_{i,k+1}^{\mathrm{T}}, \qquad (20)$$

where the square-root information matrix of dimension $(n \times m)$ is defined as

$$S_{i,k+1} = Y_{k+1|k} P_{xz,k+1|k} \overline{S}_{R,k+1} \qquad (21)$$

$$= S_{y,k+1|k} S_{y,k+1|k}^{\mathrm{T}} P_{xz,k+1|k} \overline{S}_{R,k+1}. \qquad (22)$$

Because $P_{xz,k+1|k} = T_{21} T_{11}^{\mathrm{T}}$ (see Equation (37) of [7]), we finally write (22) as

$$\boxed{S_{i,k+1} = S_{y,k+1|k} S_{y,k+1|k}^{\mathrm{T}} T_{21} T_{11}^{\mathrm{T}} \overline{S}_{R,k+1}} \qquad (23)$$

To determining i_{k+1}, we rewrite (4) as

$$\hat{y}_{k+1|k} = Y_{k+1|k} \hat{x}_{k+1|k} \qquad (24)$$

Because $Y_{k+1|k}$ is a symmetric matrix, we may also write (24) as

$$\hat{y}_{k+1|k} = Y_{k+1|k}^{\mathrm{T}} \hat{x}_{k+1|k} \qquad (25)$$

Substituting (25) into (8) yields

$$\begin{aligned} i_{k+1} &= Y_{k+1|k} P_{xz,k+1|k} R_{k+1}^{-1} \\ &\quad \left(z_{k+1} - \hat{z}_{k+1|k} + P_{xz,k+1|k}^{\mathrm{T}} Y_{k+1|k}^{\mathrm{T}} \hat{x}_{k+1|k}\right) \\ &= Y_{k+1|k} P_{xz,k+1|k} \overline{S}_{R,k+1} \overline{S}_{R,k+1}^{\mathrm{T}} \\ &\quad \left(z_{k+1} - \hat{z}_{k+1|k} + P_{xz,k+1|k}^{\mathrm{T}} Y_{k+1|k}^{\mathrm{T}} \hat{x}_{k+1|k}\right) \end{aligned} \qquad (26)$$

Substituting (21) into (26) and expanding the right hand side yields

$$\boxed{i_{k+1} = S_{i,k+1} \overline{S}_{R,k+1} \left(z_{k+1} - \hat{z}_{k+1|k}\right) + S_{i,k+1} S_{i,k+1}^{\mathrm{T}} \hat{x}_{k+1|k}} \qquad (27)$$

Consider Path 3 now. The updated information vector $\hat{y}_{k+1|k+1}$ can be computed by substituting (27) into (6). To obtain the updated information matrix $Y_{k+1|k+1}$, we replace the right hand side of with square-roots and write

$$\begin{aligned} Y_{k+1|k+1} &= S_{y,k+1|k} S_{y,k+1|k}^{\mathrm{T}} + S_{i,k+1} S_{i,k+1}^{\mathrm{T}} \\ &= \begin{bmatrix} S_{y,k+1|k} & S_{i,k+1} \end{bmatrix} \begin{bmatrix} S_{y,k+1|k} & S_{i,k+1} \end{bmatrix}^{\mathrm{T}} \end{aligned}$$

Hence, the square-root of the updated information matrix is given by

$$\boxed{S_{y,k+1|k+1} = \mathrm{Tria}\left(\begin{bmatrix} S_{y,k+1|k} & S_{i,k+1} \end{bmatrix}\right)} \qquad (28)$$

Note: When the local sensors employ the square-root information filtering algorithm, they are required to send the square-root information matrices to the fusion center (see **Figure 1**). For a distributed sensor network with n_s sensors, we may obtain $S_{y,k+1|k+1}$ by augmenting the arguments on the right hand side of (28) with n_s square-root information contribution matrices coming from n_s sensors.

$$S_{y,k+1|k+1} = \mathrm{Tria}\left(\begin{bmatrix} S_{y,k+1|k} & S_{i,k+1}^{(1)} & \cdots & S_{i,k+1}^{(n_s)} \end{bmatrix}\right). \qquad (29)$$

Table 1 below summarizes the steps involved in the SCIF algorithm.

Table 1. SCIF algorithm

SCIF: Time Update

1) Assume at time k that $\left(\hat{\boldsymbol{y}}_{k|k}, \boldsymbol{S}_{y,k|k}\right)$ is known. Compute the square-root covariance matrix

$$\boldsymbol{S}_{k|k} = \boldsymbol{S}_{y,k|k}^{-\mathrm{T}} \tag{30}$$

2) Compute the state estimate

$$\hat{\boldsymbol{x}}_{k|k} = \boldsymbol{S}_{k|k}\boldsymbol{S}_{k|k}^{\mathrm{T}}\hat{\boldsymbol{y}}_{k|k} \tag{31}$$

3) Use the time-update of the SCKF to compute $\left(\hat{\boldsymbol{x}}_{k+1|k}, \boldsymbol{S}_{k+1|k}\right)$ from $\left(\hat{\boldsymbol{x}}_{k|k}, \boldsymbol{S}_{k|k}\right)$.

4) Compute the square-root of the predicted information matrix

$$\boldsymbol{S}_{y,k+1|k} = \boldsymbol{S}_{k+1|k}^{-\mathrm{T}} \tag{32}$$

5) Compute the predicted information vector

$$\hat{\boldsymbol{y}}_{k+1|k} = \boldsymbol{S}_{y,k+1|k}\boldsymbol{S}_{y,k+1|k}^{\mathrm{T}}\hat{\boldsymbol{x}}_{k+1|k} \tag{33}$$

SCIF: Measurement Update

1) Use the measurement-update of the SCKF to compute $\boldsymbol{T}_{11}$ and $\boldsymbol{T}_{21}$ from $\left(\hat{\boldsymbol{x}}_{k+1|k}, \boldsymbol{S}_{k+1|k}\right)$.

2) Compute the square-root of the information contribution matrix

$$\boldsymbol{S}_{i,k+1} = \boldsymbol{S}_{y,k+1|k}\boldsymbol{S}_{y,k+1|k}^{\mathrm{T}}\boldsymbol{T}_{21}\boldsymbol{T}_{11}^{\mathrm{T}}\overline{\boldsymbol{S}}_{R,k+1}, \tag{34}$$

where the inverse of the measurement noise covariance matrix

$$\overline{\boldsymbol{S}}_{R,k+1} = \boldsymbol{S}_{R,k+1}^{-\mathrm{T}}. \tag{35}$$

3) Compute the information contribution vector

$$\boldsymbol{i}_{k+1} = \boldsymbol{S}_{i,k+1}\overline{\boldsymbol{S}}_{R,k+1}\left(\boldsymbol{z}_{k+1} - \hat{\boldsymbol{z}}_{k+1|k}\right) + \boldsymbol{S}_{i,k+1}\boldsymbol{S}_{i,k+1}^{\mathrm{T}}\hat{\boldsymbol{x}}_{k+1|k}. \tag{36}$$

4) Compute the updated information vector

$$\hat{\boldsymbol{y}}_{k+1|k+1} = \hat{\boldsymbol{y}}_{k+1|k} + \boldsymbol{i}_{k+1} \tag{37}$$

5) Compute the square-root of the updated information matrix

$$\boldsymbol{S}_{y,k+1|k+1} = \mathrm{Tria}\left(\begin{bmatrix} \boldsymbol{S}_{y,k+1|k} & \boldsymbol{S}_{i,k+1} \end{bmatrix}\right). \tag{38}$$

4. Application to Decentralized Sensor Fusion

Context. We consider an air-traffic control scenario, where an aircraft executes maneuvering turn in a horizontal plane at a constant, but unknown turn rate Ω. **Figure 4** shows a representative trajectory of the aircraft. The kinematics of the turning motion can be modeled by the following nonlinear process equation [1]:

$$\boldsymbol{x}_k = \begin{pmatrix} 1 & \dfrac{\sin\Omega T}{\Omega} & 0 & -\left(\dfrac{1-\cos\Omega T}{\Omega}\right) & 0 \\ 0 & \cos\Omega T & 0 & -\sin\Omega T & 0 \\ 0 & \dfrac{1-\cos\Omega T}{\Omega} & 1 & \dfrac{\sin\Omega T}{\Omega} & 0 \\ 0 & \sin\Omega T & 0 & \cos\Omega T & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}\boldsymbol{x}_{k-1} + \boldsymbol{v}_{k-1},$$

where the state of the aircraft
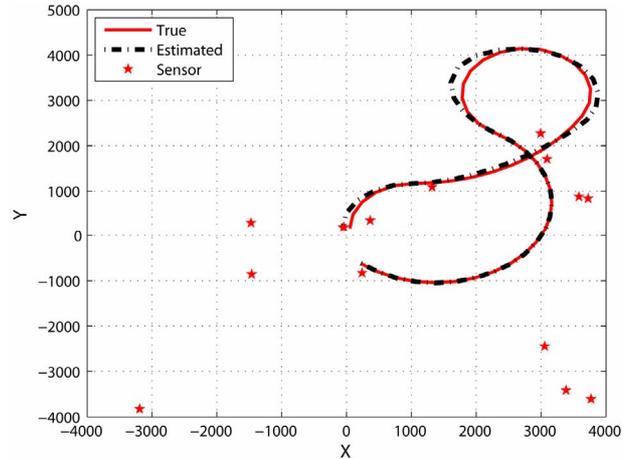
Figure 4. True aircraft trajectory—solid line, SCIF estimate—dotted line, $\star$—Radar locations.

$x = [\boldsymbol{x}[1]\ \boldsymbol{x}[2]\ \boldsymbol{x}[3]\ \boldsymbol{x}[4]\ \boldsymbol{x}[5] = \Omega]^{\mathrm{T}}$; $\boldsymbol{x}[1]$ and $\boldsymbol{x}[3]$ denote positions, and $\boldsymbol{x}[2]$ and $\boldsymbol{x}[4]$ denote velocities

in the x and y directions, respectively; T is the time-interval between two consecutive measurements; the process noise $v_{k-1} \sim \mathcal{N}(0, Q)$ with a nonsingular covariance $Q = \text{diag}[q_1 M \quad q_1 M \quad q_2 T]$, where

$$M = \begin{pmatrix} \dfrac{T^3}{3} & \dfrac{T^2}{2} \\ \dfrac{T^2}{2} & T \end{pmatrix};$$

The scalar parameters q_1 and q_2 are related to process noise intensities. The radars were assumed to measure range and range-rate. For a radar located at $x_s = [x_s[1], x_s[2]]^T$, the measurement equation is given by

$$z_k = \begin{pmatrix} \sqrt{(x_k[1] - x_s[1])^2 + (x_k[3] - x_s[2])^2} \\ \dfrac{(x_k[1] - x_s[1])x_k[2] + (x_k[3] - x_s[2])x_k[4]}{\sqrt{(x_k[1] - x_s[1])^2 + (x_k[3] - x_s[2])^2}} \end{pmatrix} + w_k$$

(39)

where the measurement noise covariance R is given by

$$R = \text{cov}[w_k] = \text{diag}\left(\begin{bmatrix} \sigma_r^2 & \sigma_{\dot{r}}^2 \end{bmatrix} \right).$$

To make this nonlinear tracking problem highly difficult, the target trajectory was made up of four segments, in each of which Ω was set to be $5° \, \text{s}^{-1}, -9° \, \text{s}^{-1}, -3° \, \text{s}^{-1}$ and $9° \, \text{s}^{-1}$ for the duration of 0 - 40 s, 40 - 70 s, 70 - 90 s and 90 - 100 s, respectively. We used the following parameters for simulation:

$$T = 2 \, \text{s}$$

$$q_1 = 0.1 \, \text{m}^2 \cdot \text{s}^{-3}$$

$$q_2 = 10^{-6} \, \text{s}^{-3}$$

$$\sigma_r = 10 \, \text{m}$$

$$\sigma_{\dot{r}} = 10 \, \text{m/s}$$

The true initial state was assumed to be at

$$x_0 = \begin{bmatrix} 0 \, \text{m} & 100 \, \text{m} \cdot \text{s}^{-1} & -400 \, \text{m} & 120 \, \text{m} \cdot \text{s}^{-1} & 2° \, \text{s}^{-1} \end{bmatrix}^T$$

The initial state estimate $\hat{x}_{0/0}$ was chosen randomly from $\mathcal{N}(x_0, P_{0/0})$ in each run, where

$$P_{0/0} = \text{diag}[100 \, \text{m}^2 \quad 25 \, \text{m}^2 \cdot \text{s}^{-2} \quad 25 \, \text{m}^2 \quad 25 \, \text{m} \cdot \text{s}^{-2}$$
$$\left(1.7 \, \text{m} \cdot \text{rad} \cdot \text{s}^{-1}\right)^2].$$

For a fair evaluation, we made a total of $N = 100$ independent Monte Carlo runs (The reader may refer to https://sites.google.com/site/haranarasaratnam/software for a set of Matlab code used to generate the results). The radars were randomly placed in a square-shaped surveillance region with the opposite vertices at (−4000, −4000)

and (4000, 4000). In this experiment, the number of radars, n_s, was varied from one to fifteen. We employed the following information filters for tracking the aircraft:

- Extended Information Filter (EIF);
- Cubature Information Filter (CIF);
- Square-root Cubature Information Filter (SCIF).

Note that the unscented information filter boils down to the CIF when the free parameter κ is forced to take zero. For this reason, the unscented information filter was excluded from comparison.

Performance Metrics. For performance comparison, we computed the accumulative root mean-squared error (ARMSE) in position and velocity. The ARMSE yields a combined measure of the bias and variance of a filter estimate. We define the ARMSE in position

$$\text{ARMSE}[\text{pos}]$$

$$= \sqrt{\frac{1}{N} \sum_{n=1}^{N} \frac{1}{K} \sum_{k=1}^{K} \left(x_{k,n}[1] - \hat{x}_{k|k,n}[1]\right)^2 + \left(x_{k,n}[3] - \hat{x}_{k|k,n}[3]\right)^2}$$

where $\left(x_{k,n}[1], x_{k,n}[3]\right)$ and $\left(\hat{x}_{k|k,n}[1], \hat{x}_{k|k,n}[3]\right)$ are the true and globally estimated positions at the n-th Monte Carlo run. Similarly to the ARMSE in position, we may also write formula of the ARMSE in velocity. To check the numerical robustness of an information filter, the filter divergence rate was introduced. The filter was declared to diverge when $\sqrt{\text{MSE}[\text{pos}]}$ of a specific Monte Carlo run exceeded 100 m. Subsequently, those diverged runs were excluded from the final calculations of $\text{ARMSE}[\text{pos}]$ and $\text{ARMSE}[\text{vel}]$.

Observations. **Figures 5(a)** and **(b)** show the ARMSE in position and velocity, respectively, for the SCIF and EIF. As expected, as the number of sensors n_s increases, the ARMSEs in position and velocity decrease. However, the performance gain quickly diminishes; as $n_s > 5$, the performance gain seems to be trivial. Moreover, the SCIF consistently outperforms the EIF irrespective of n_s although the performance deviation between the two reduces with n_s. The CIF is not included in **Figures 5(a)** and **(b)** because its performance was almost identical to the SCIF.

Figure 6 shows how many times each filter experienced divergence out of $N = 100$ independent Monte Carlo runs. As can be seen from **Figure 6**, the EIF diverges the most. However, its divergence rate decreases as n_s increases. The CIF diverges a few times especially when n_s is less. The SCIF diverges a few times only when $n_s = 2$. Unlike the CIF, the SCIF does not diverge when $n_s = 3$ and 4. This observation indicates that the SCIF is superior to other filters in terms of numerical robustness. None of the filters diverged when $n_s \geq 7$.

5. Concluding Remarks

This paper has presented a numerically robust square-root
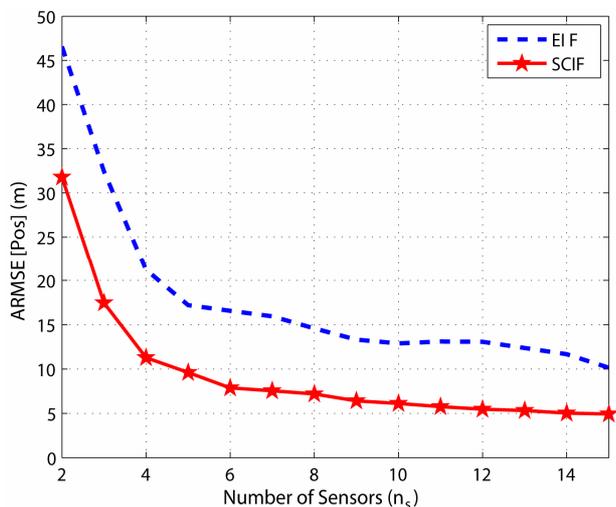
(a) Position Error (m)
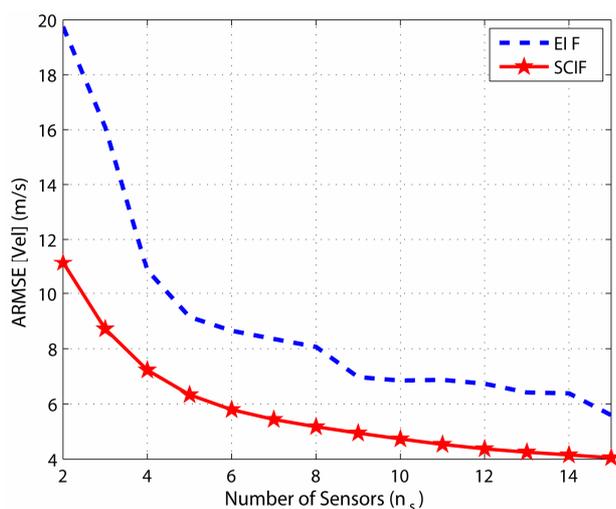
(b) Velocity Error (m/s)

Figure 5. Accumulative root mean-squared error.
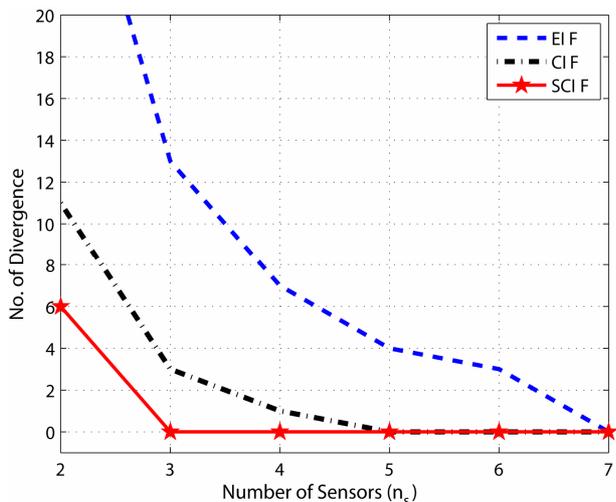
Figure 6. Number of divergence experienced in 100 independent Monte Carlo runs.

cubature information filter (SCIF) algorithm for multi-sensor fusion in a nonlinear Gaussian environment. Because the SCIF propagates the square-root matrices, it avoids to compute numerically sensitive matrix calculations. It was successfully demonstrated that the proposed SCIF algorithm is numerically robust using a computer experiment in which a target tracking problem in a distributed sensor network with feedback was considered. One of the interesting future research topics is to compare the computational demands of various information filters.

REFERENCES

[1] Y. Bar Shalom, X. R. Li and T. Kirubarajan, "Estimation with Applications to Tracking and Navigation," Wiley & Sons, New York, 2001. doi:10.1002/0471221279

[2] Y. Bar-Shalom and X. R. Li, "Multitarget-Multisensor Tracking: Principles and Techniques," YBS, Storrs, 1995.

[3] Y. Zhu, Z. You, J. Zhao, K. Zhang and X. Li, "The Optimality for the Distributed Kalman Filtering Fusion," *Automatica*, Vol. 37, No. 9, 2001, pp. 1489-1493. doi:10.1016/S0005-1098(01)00074-7

[4] T. Vercauteren and X. Wang, "Decentralized Sigma-Point Information Filters for Target Tracking in Collaborative Sensor Networks," *IEEE Transactions on Signal Processing*, Vol. 53, No. 8, 2005, pp. 2997-3009. doi:10.1109/TSP.2005.851106

[5] Y. Kim, J. Lee, H. Do, B. Kim, T. Tanikawa, K. Ohba, G. Lee and J. Yun, "Unscented Information Filtering Method for Reducing Multiple Sensor Registration Error," *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Seoul, 20-22 August 2008, pp. 326-331. doi:10.1109/MFI.2008.4648086

[6] I. Arasaratnam and S. Haykin, "Cubature Kalman Filters," *IEEE Transactions on Automatic Control*, Vol. 54, No. 6, 2009, pp. 1254-1269.

[7] I. Arasaratnam, S. Haykin and T. R. Hurd, "Cubature Kalman Filtering for Continuous-Discrete Systems: Theory and Simulations," *IEEE Transactions on Signal Processing*, Vol. 58, No. 10, 2010, pp. 4977-4993.

[8] G. H. Golub and C. F. Van Loan, "Matrix Computations," MD John Hopkins University Press, Baltimore, 1996.