The Square-Root Sigma-Point Information Filters for Nonlinear Estimation and Sensor Fusion

Guoliang Liu, *Graduate Student Member, IEEE,* Florentin Wörgötter and Irene Markelić

***Abstract*—The unscented information filter (UIF) has been introduced recently for nonlinear system estimation and sensor fusion. In the UIF framework, a number of sigma points are sampled from the probability distribution of the prior state by the unscented transform and then propagated through the nonlinear dynamic function and measurement function. The new state is estimated from the propagated sigma points. In this way, the UIF can achieve higher estimation accuracies and faster convergence rates than the extended information filter (EIF), which uses a Taylor series to linearize the nonlinear function. This paper extends the framework of the UIF: first, a central difference information filter (CDIF) is derived by employing Stirling's interpolation to generate the sigma points. This leads to fewer predefined parameters and lower computational cost as compared to the original UIF. Second, we introduce the square-root forms of the CDIF and UIF to increase the numerical stability and guarantee positive semi-definiteness of the state covariances. The proposed algorithms are finally evaluated on two nonlinear problems: the classical space-vehicle tracking problem and the bearing-only tracking problem.**

***Index Terms*—Nonlinear estimation, multiple sensor fusion, target tracking, sigma point filters, central difference information filter, square-root filters.**

I. Introduction

THE accuracy and robustness of control systems can be improved using fused information from multiple sensors. Therefore, sensor fusion techniques have been widely studied in many research fields, i.e., robot navigation, surveillance, and intelligent vehicles [1]. Recently, the information filter (IF), which is the dual of the Kalman filter (KF), has attracted much attention for multiple sensor fusion [2]. Both the IF and the KF represent distributions of random state variables with Gaussians. However, in contrast to moment parametrization as done in the KF, the IF uses an information matrix and an information vector to represent the Gaussians. This difference in parameterization makes the IF superior to the KF concerning multiple sensor fusion, as computations are simpler and no prior information of the system state is required [3].

In the case of nonlinear estimation problems, an extended version of the IF can be obtained using the first order term of the Taylor series expansions of the nonlinear functions, i.e., the dynamic and measurement functions of the system, which is called extended information filter (EIF). This approximation can introduce large errors when the system model is highly nonlinear, and the higher order terms of Taylor series are important [4]. To address this issue, the unscented information filter (UIF) has been proposed by Kim *et al.* [5] and Lee [3]. Kim developed the UIF by using minimum mean square error estimation. In contrast, Lee's UIF algorithm is derived by embedding statistical linear error propagation into the EIF architecture. Although their methods are different, results are essentially identical [3], [5], [6]. The UIF uses a number of deterministic sigma points to capture the true information matrix and the information vector, which can be accurate up to the second order of any nonlinearity. However, three parameters (α, β, κ) must be defined for the UIF, which depend on the system models. As shown in [3], [7], the UIF is superior to the EIF not only in terms of estimation accuracy but also concerning the convergence speed for nonlinear estimation and multiple sensor fusion. However, the choice of system parameters (α, β, κ) can affect the filter's estimation precision.

In this paper, we first propose an alternative to the UIF, which we call *central difference information filter* (CDIF). Where the UIF uses the unscented transform to compute the sigma points, the CDIF employs Stirling's interpolation. As proved in [4], Stirling's interpolation based central difference Kalman filter (CDKF) has the same or superior performance as the unscented transform based Kalman filter (UKF), with one advantage over the UKF: Stirling's interpolation only needs a single parameter, the interval size h, whereas the unscented transform needs three [8]. As shown in our simulation experiments, the CDIF not only inherits the simplicity of the IF for multiple sensor fusion (while having the same accuracy as the UIF), it also has a lower computational cost.

Second, we propose to use square-root forms for both the UIF and the CDIF. In Stirling's interpolation and the unscented transform, a square-root of the prior covariance has to be calculated to generate sigma points. This step is computationally expensive and requires that the covariance matrix to be positive definite. To save computational cost and increase numerical robustness, a square-root form of the covariance is directly taken and updated in the algorithm. This idea has been introduced by Van der Merwe [9], who proposed square-root forms of Kalman filters. Here we employ a similar idea, and describe the square-root versions of UIF and CDIF for solving nonlinear state estimation and sensor fusion problems.

This paper is organized as follows: First, we present our CDIF algorithm for nonlinear estimation and multiple sensor fusion in Section II, and then the Square-Root CDIF and Square-Root UIF are proposed in Section III and Section IV respectively. Simulation results of target tracking are presented and discussed in Section V. Finally, the work is concluded in Section VI.

Guoliang Liu, Florentin Wörgötter and Irene Markelić are with Bernstein Center for Computational Neuroscience, III Physikalisches Institut - Biophysik, University of Göttingen, 37077 Germany (e-mail: {liu, worgott, irene}@physik3.gwdg.de)

II. Central Difference Information filter

In this section, we present our CDIF framework, which replaces the unscented transform with Stirling's interpolation to generate the sigma points. The algorithm includes three steps: prediction, measurement update and global information fusion.

A. Stirling's interpolation

Stirling's interpolation has been used previously with the Kalman filter in the literature, referred to as the *central difference Kalman filter* (CDKF) [4], [8]. The CDKF uses a symmetric set of $2L+1$ sigma points to approximate nonlinear functions. In the case of Gaussian distributions of the system variables, the mean and covariance can be represented by those sigma points. As we mentioned in Section I, the IF is a dual filter of the KF, such that the information vector and matrix also can be derived by those sigma points. In this section, we first show how the mean and covariance are derived using Stirling's interpolation, then show how the information vector and matrix are obtained from the mean and covariance.

The $2L+1$ prior sigma points used in Stirling's interpolation step are given by the prior mean $\hat{x}$ plus or minus the columns of the scaled square root of the prior covariance matrix P_x [4]:

$$\mathscr{X}_i = \begin{cases} \hat{x}, & i=0 \\ \hat{x}+(h\sqrt{P_x})_i, & i=1,\cdots,L \\ \hat{x}-(h\sqrt{P_x})_i, & i=L+1,\cdots,2L \end{cases} \tag{1}$$

where h is a scaling parameter and L is the dimension of the state $\hat{x}$. The subscript i indicates the ith column of the matrix. A set of the posterior sigma points can be derived by propagating these prior sigma points through the nonlinear function g: $\mathscr{Z}_i = g(\chi_i)$. Furthermore, the estimations of mean $\hat{z}$, covariance P_z and cross-covariance P_{xz} are obtained as follows:

$$\bar{z} \approx \sum_{i=0}^{2L} w_i^{(m)} \mathscr{Z}_i \tag{2}$$

$$\begin{aligned} P_z \approx & \sum_{i=1}^{L} w_i^{(c_1)} (\mathscr{Z}_i - \mathscr{Z}_{i+L})(\mathscr{Z}_i - \mathscr{Z}_{i+L})^T \\ & + \sum_{i=1}^{L} w_i^{(c_2)} (\mathscr{Z}_i + \mathscr{Z}_{i+L} - 2\mathscr{Z}_0)(\mathscr{Z}_i + \mathscr{Z}_{i+L} - 2\mathscr{Z}_0)^T \end{aligned} \tag{3}$$

$$P_{xz} \approx \sqrt{w_1^{(c1)}} P_x (\mathscr{Z}_{1:L} - \mathscr{Z}_{L+1:2L})^T. \tag{4}$$

The corresponding weights for the mean and covariance are defined as

$$\begin{aligned} w_0^{(m)} &= \tfrac{h^2-L}{h^2} \\ w_i^{(m)} &= \tfrac{1}{2h^2}, \\ w_i^{(c1)} &= \tfrac{1}{4h^2}, \\ w_i^{(c2)} &= \tfrac{h^2-1}{4h^4}, \quad i=1,\cdots,2L \end{aligned} \tag{5}$$

As proved in [4], if the random variables obey a Gaussian distribution, the optimal value of h is $\sqrt{3}$. Stirling's interpolation only depends on one parameter, the interval size h, in contrast to three parameters (α,β,κ) which are required in the unscented transform. This makes Stirling's method simpler and easier to tune.

B. Prediction

Here we consider the discrete-time nonlinear dynamic system

$$x_k = f(x_{k-1}, v), \tag{6}$$

where x_k is the state vector of the system at time step k, and $v \sim \mathscr{N}(\bar{v}, R_v)$ is Gaussian noise.

First, the state vector is augmented with the noise variable and the corresponding augmented covariance matrix is derived by:

$$x_{k-1}^{a_v} = \begin{bmatrix} x_{k-1} \\ \bar{v} \end{bmatrix}, P_{k-1}^{a_v} = \begin{bmatrix} P_{k-1} & 0 \\ 0 & R_v \end{bmatrix}. \tag{7}$$

A symmetric set of $2L+1$ sigma points is generated using (1):

$$\mathscr{X}_{i,k-1}^{a_v} = \begin{cases} x_{k-1}^{a_v}, & i=0 \\ x_{k-1}^{a_v} + (h\sqrt{P_{k-1}^{a_v}})_i, & i=1,\cdots,L \\ x_{k-1}^{a_v} - (h\sqrt{P_{k-1}^{a_v}})_i, & i=L+1,\cdots,2L \end{cases} \tag{8}$$

where each sigma point $\mathscr{X}_{i,k-1}^{a_v}$ contains the state and noise variable components

$$\mathscr{X}_{i,k-1}^{a_v} = \begin{bmatrix} \mathscr{X}_{i,k-1}^{x} \\ \mathscr{X}_{i,k-1}^{v} \end{bmatrix}. \tag{9}$$

These sigma points are further passed through the nonlinear function (6), such that the predicted sigma points for the discrete time k are derived as:

$$\mathscr{X}_{i,k|k-1}^{x} = f(\mathscr{X}_{i,k-1}^{x}, \mathscr{X}_{i,k-1}^{v}). \tag{10}$$

Finally, the first two moments of the predicted state vector are obtained by linear regression of the transformed sigma points:

$$x_{k|k-1}^{-} = \sum_{i=0}^{2L} w_i^m \mathscr{X}_{i,k|k-1}^{x} \tag{11}$$

$$P_{k|k-1}^{-} = \sum_{i=1}^{L} w_i^{(c_1)} \alpha_i \alpha_i^T + \sum_{i=1}^{L} w_i^{(c_2)} \beta_i \beta_i^T, \tag{12}$$

where $\alpha_i = \mathscr{X}_{i,k|k-1}^{x} - \mathscr{X}_{i+L,k|k-1}^{x}$ and $\beta_i = \mathscr{X}_{i,k|k-1}^{x} + \mathscr{X}_{i+L,k|k-1}^{x} - 2\mathscr{X}_{0,k|k-1}^{x}$. As stated in Section I, the information matrix and information vector are the dual of the mean and covariance, so that the predicted information matrix $Y_{k|k-1}$ and the information vector $y_{k|k-1}$ are derived as:

$$y_{k|k-1} = Y_{k|k-1}\, x_{k|k-1}^{-} \tag{13}$$

$$Y_{k|k-1} = (P_{k|k-1}^{-})^{-1}. \tag{14}$$

C. Measurement update

The measurement function of the nonlinear system is defined as

$$z_k = h(x_k) + n, \tag{15}$$

where z_k is the measurement and $n \sim \mathscr{N}(\bar{n}, R_n)$ is the Gaussian noise of the measurement.

The sigma points used for the measurement update are derived as:

$$\mathscr{X}_{i,k|k-1} = \begin{cases} x_{k|k-1}^{-}, & i=0 \\ x_{k|k-1}^{-} + (h\sqrt{P_{k|k-1}^{-}})_i, & i=1,\cdots,L \\ x_{k|k-1}^{-} - (h\sqrt{P_{k|k-1}^{-}})_i, & i=L+1,\cdots,2L \end{cases} \tag{16}$$

The predicted measurement points are obtained by transforming the sigma points through (15)

$$\mathscr{Z}_{i,k|k-1} = h(\mathscr{X}_{i,k|k-1}). \tag{17}$$

Furthermore, the mean and cross-covariance are derived by:

$$z^-_{k|k-1} = \sum_{i=0}^{2L} w_i^m \mathscr{Z}_{i,k|k-1} \tag{18}$$

$$P^{xz}_{k|k-1} = \sqrt{w_1^{(c1)} P_{k|k-1}}(\mathscr{Z}_{1:L} - \mathscr{Z}_{L+1:2L})^T. \tag{19}$$

Finally, the measurement update of the information vector and the information matrix are derived as:

$$y_k = y_{k|k-1} + \phi_k \tag{20}$$

$$Y_k = Y_{k|k-1} + \Phi_k \tag{21}$$

where ϕ_k and Φ_k are information contribution terms for the information vector and matrix respectively, which can be derived by:

$$\phi_k = Y_{k|k-1} P^{xz}_{k|k-1} R_n^{-1}[z_k - z^-_{k|k-1} + (P^{xz}_{k|k-1})^T y_{k|k-1}] \tag{22}$$

$$\Phi_k = Y_{k|k-1} P^{xz}_{k|k-1} R_n^{-1} (P^{xz}_{k|k-1})^T (Y_{k|k-1})^T. \tag{23}$$

The derivation of (22) and (23) can be found in [3], [5], [7].

D. Global information fusion

In case of multiple sensors N, where the measurement noises between the sensors are uncorrelated, the measurement update for information fusion is simply expressed as a linear combination of the local information contribution terms:

$$y_k = y_{k|k-1} + \sum_{i=1}^{N} \phi_{i,k} \tag{24}$$

$$Y_k = Y_{k|k-1} + \sum_{i=1}^{N} \Phi_{i,k}. \tag{25}$$

III. Square-Root CDIF

For each discrete time update in the CDIF, a new set of sigma points needs to be calculated, which is computationally expensive because the square root of the covariance matrix is required. To avoid this operation, we introduce a Square-Root CDIF (SRCDIF). This square root form guarantees that the state covariance matrix P is positive semi-definite, and further improves the numerical stability of the system [9].

A. Square-Root CDIF for state estimation

The square-root CDIF benefits from three powerful matrix factorization techniques: *QR decomposition*, *Cholesky factor updating* and *efficient least squares*. In the following, we will use *qr*, *chol*, *cholupdate* and '\' (backslash) to refer to the *QR decomposition*, *Cholesky decomposition*, *Cholesky factor updating* and *efficient least squares* respectively [1].

[1]The abbreviations *qr*, *chol*, *cholupdate* and '\' (backslash) are in accordance with the function names for *QR decomposition*, *Cholesky decomposition*, *Cholesky factor updating* and *efficient least squares* in Matlab.

- *QR decomposition*. In the CDIF, the square-root of the covariance matrix S is derived by *Cholesky decomposition* on P: $S = chol(P)^T$ where S is a lower triangular matrix and fulfills $P = SS^T$. If we know $P = AA^T$, the square-root factor S can be directly calculated from A by *QR decomposition*: $S = qr(A)^T$. If the matrix $A \in \mathbb{R}^{L\times N}$, then the computational complexity of a *QR decomposition* is $\mathscr{O}(NL^2)$.
- *Cholesky factor updating*. If the original update of the covariance matrix is $P \pm uu^T$ and S is the Cholesky factor, then the rank 1 update of S is $S = cholupdate(S, u, \pm)$ where u is the update vector. If u is a matrix, we can update each column of u one by one in a loop. For each column vector, the computational complexity is $\mathscr{O}(L^2)$. This procedure can alternatively be implemented as $S = qr([S \ \pm u]^T)$ using *QR decomposition* without the loop updates.
- *Efficient least squares*. The least squares solution for the linear equation $Px = b$ can be solved efficiently using forward and back substitution if the Cholesky factor S is known and satisfies $P = SS^T$. For example, we can solve it by $x = S^T \backslash (S \backslash b)$ where '\' is the backslash. This operation only requires computational complexity $\mathscr{O}(L^2)$.

The whole process is shown in Algorithm 1, where h is the scaling parameter derived from (1), L is the dimension of the state, R_v and R_n are process noise covariance and observation noise covariance respectively, $w_i^{(m)}$ and $w_i^{(c)}$ are weights calculated in (5), and I is an identity matrix. Algorithm 1 looks similar to the general CDIF algorithm introduced in Section II, except that the *Cholesky factor* S_{x_k} is used instead of the covariance P_k. The Square-Root CDIF also comprises two steps, the first is the prediction and the second is the measurement update. For each step, a number of sigma points are generated using *Stirling's interpolation* in (27) and (35). However, the square root of covariance S_{x_k} is directly used to calculate the sigma points without the *Cholesky factorization*.

In the prediction step, the *Cholesky factor* S_{x_k} is updated using *QR decomposition* on the weighted sigma points. This step replaces the P_k update in (12) and has complexity $\mathscr{O}(L^3)$. The information vector $\hat{y}^-_k = (P^-_k)^{-1}\hat{x}^-_{x_k} = \left(S^-_{x_k}\right)^T \backslash \left(S^-_{x_k} \backslash \hat{x}^-_k\right)$ is derived using *efficient least squares* in (33). Because $\hat{S}^-_{x_k}$ is a square and triangular matrix, we can directly use back-substitution for solving $\hat{y}^-_k$ without the need for matrix inversion. The back substitution only requires $\mathscr{O}(L^2)$. Next is the calculation of the square-root information matrix $S^-_{y_k}$ in (34). This step requires a *QR decomposition* since $S^-_{y_k}$ is a upper triangular matrix, and meets $\left(S^-_{y_k}\right)^T S^-_{y_k} = \left(S^-_{x_k}\right)^{-T}\left(S^-_{x_k}\right)^{-1}$. As $S^-_{x_k}$ is a lower triangular matrix, *QR decomposition* is used to solve the *Cholesky factor* $S^-_{y_k}$ of the information matrix $Y_{k|k-1}$. To avoid the inversion, here we use *efficient least squares* to solve $\left(S^-_{x_k}\right)^{-1}$ as $S^-_{x_k} \backslash I$, where I is an identity matrix.

In the measurement update step, the updated information vector y_k is derived by *efficient least squares* in (40). If the observation dimension is M, the updated square-root information matrix S_{y_k} is calculated in (41) by applying an M-sequential Cholesky update to $S^-_{y_k}$. The columns of matrix U are update vectors. This requires $\mathscr{O}(L^2M)$ and replaces the measurement

Algorithm 1 Square-Root CDIF for state estimation

- Initialization:
  $\hat{x}_0 = E(x_0)$, $S_{x0} = chol\left\{E\left((x_0-\hat{x}_0)(x_0-\hat{x}_0)^T\right)\right\}$, $S_v = \sqrt{R_v}$ and $S_n = \sqrt{R_n}$.
- For $k = 1, \cdots, \infty$:
  1) Generate sigma points for prediction:

$$\hat{x}^{a_v}_{k-1} = \begin{bmatrix}\hat{x}_{k-1}\\ \bar{v}\end{bmatrix}, \quad S^{a_v}_{k-1} = \begin{bmatrix}S_{x_{k-1}} & 0\\ 0 & S_v\end{bmatrix} \tag{26}$$

$$\mathscr{X}^{a_v}_{k-1} = \begin{bmatrix}\hat{x}^{a_v}_{k-1} & \hat{x}^{a_v}_{k-1}+hS^{a_v}_{k-1} & \hat{x}^{a_v}_{k-1}-hS^{a_v}_{k-1}\end{bmatrix} \tag{27}$$

  2) Prediction equations:

$$\mathscr{X}^{x}_{k|k-1} = f(\mathscr{X}^{x}_{k-1}, \mathscr{X}^{v}_{k-1}, u_{k-1}) \tag{28}$$

$$\hat{x}^{-}_{k} = \sum_{i=0}^{2L} w_i^{(m)}\, \mathscr{X}^{x}_{i,k|k-1} \tag{29}$$

$$A = \sqrt{w_1^{(c_1)}}\left(\mathscr{X}^{x}_{1:L,k|k-1} - \mathscr{X}^{x}_{L+1:2L,k|k-1}\right) \tag{30}$$

$$B = \sqrt{w_1^{(c_2)}}(\mathscr{X}^{x}_{1:L,k|k-1} + \mathscr{X}^{x}_{L+1:2L,k|k-1} - 2\mathscr{X}^{x}_{0,k|k-1}) \tag{31}$$

$$S^{-}_{x_k} = qr\{[A \quad B]\} \tag{32}$$

$$\hat{y}^{-}_{k} = \left(S^{-}_{x_k}\right)^T \backslash \left(S^{-}_{x_k} \backslash \hat{x}^{-}_{k}\right) \tag{33}$$

$$S^{-}_{y_k} = qr\left\{S^{-}_{x_k} \backslash I\right\} \tag{34}$$

  3) Generate sigma points for measurement update:

$$\mathscr{X}_{k|k-1} = \begin{bmatrix}\hat{x}^{-}_{k} & \hat{x}^{-}_{k}+hS^{-}_{x_k} & \hat{x}^{-}_{k}-hS^{-}_{x_k}\end{bmatrix} \tag{35}$$

  4) Measurement update equations:

$$\mathscr{Z}_{k|k-1} = h\left(\mathscr{X}_{k|k-1}\right) \tag{36}$$

$$\hat{z}^{-}_{k} = \sum_{i=0}^{2L} w_i^{(m)}\, \mathscr{Z}_{i,k|k-1} \tag{37}$$

$$P_{x_k z_k} = \sqrt{w_1^{c_1}} S^{-}_{x_k}[\mathscr{Z}_{1:L,k|k-1} - \mathscr{Z}_{L+1:2L,k|k-1}]^T \tag{38}$$

$$U = \left(S^{-}_{x_k}\right)^T \backslash \left(S^{-}_{x_k} \backslash P_{x_k z_k}\right)/S_n \tag{39}$$

$$y_k = \hat{y}^{-}_{k} + U/S_n^T(z_k - \hat{z}^{-}_{k} + P^T_{x_k z_k}\hat{y}^{-}_{k}) \tag{40}$$

$$S_{y_k} = cholupdate\{S^{-}_{y_k}, U, +1\} \tag{41}$$

update of Y_k in (21).

B. Square-Root CDIF for multiple sensor fusion

In the case where information from multiple sensors is available, i.e., $N > 1$, we can fuse this using the Square-Root CDIF. For the i_{th} sensor, the information contribution for the information vector is

$$\phi_{i,k} = U/S_n^T(z_k - \hat{z}^{-}_{k} + P^T_{x_k z_k}\hat{y}^{-}_{k}) \tag{42}$$

where U is defined in (39). The information contribution for the square-root information matrix is

$$S_{i,\phi_k} = U. \tag{43}$$

The final estimated result is derived by:

$$y_k = \hat{y}^{-}_{k} + \sum_{i=0}^{N} \phi_{i,k} \tag{44}$$

$$S_{y_k} = cholupdate\{S^{-}_{y_k}, [S_{1,\phi_k} \quad S_{2,\phi_k} \quad \cdots \quad S_{N,\phi_k}], +1\}. \tag{45}$$

IV. Square-Root UIF

In this section we consider the square-root implementation of the UIF. Because the UIF uses the *unscented transform* to calculate the sigma points, the architecture of the Square-Root UIF (SRUIF) has few differences from the SRCDIF. As mentioned in Section III, the main techniques behind the square-root form estimators are: *QR decomposition*, *Cholesky factor updating* and *efficient least squares*. We show how to use these in the Square-Root UIF in the following.

The Square-Root UIF is shown in Algorithm 2, where $\gamma = \sqrt{(\lambda + L)}$ is the composite scaling parameter, $\lambda = \alpha^2(L+\kappa) - L$, α and κ are scaling parameters that determine how far the sigma points spread from the mean value [4], [10], L is the dimension of the state, R_v and R_n are process noise covariance and observation noise covariance respectively, $w_i^{(m)}$ and $w_i^{(c)}$ are weights calculated by $w_0^m = \frac{\lambda}{L+\lambda}$, $w_0^c = \frac{\lambda}{L+\lambda} + (1-\alpha^2+\beta)$, $w_i^m = w_i^c = \frac{1}{2(L+\lambda)} \quad i = 1, \cdots, 2L$, and $sign\{\}$ is the *signum* function.

We compare the SRUIF in Algorithm 2 to the SRCDIF in Algorithm 1. First of all, the SRUIF uses the *unscented transform* to calculate the sigma points in (47) and (55), where the scaling parameter becomes $\gamma = \sqrt{(\lambda + L)}$ and $\lambda = \alpha^2(L+\kappa) - L$. In contrast to only one scaling parameter h used in the SRCDIF, the SRUIF depends on three parameters λ, α and κ. Second, since the weight $w_0^{(c)}$ might be negative, we need an additional *cholupdate* to update the *Cholskey factor* $S^{-}_{x_k}$ in (52), whereas the SRCDIF does not need this step. Finally, for multiple sensor fusion, the SRUIF is equivalent to the SRCDIF in (44) and (45).

V. Experiments

A. Space-vehicle tracking

To demonstrate the performance of the CDIF, UIF and their square-root forms SRCDIF and SRUIF, here we consider a classic space-vehicle reentry tracking problem, which was used in [3], [11], [12]. Two radars, which measure range and bearing, are used for tracking a high speed vehicle. The true trajectory of this vehicle is shown in Fig. 1f.

The state space of the filter consists of the position (x_1 and x_2), the velocity (x_3 and x_4) and a parameter related to the aerodynamic force x_5. As described in [11], the vehicle state dynamics for the discrete case are given by

$$\begin{aligned}
x_1(k+1) &= x_1(k) + \Delta t x_3(k)\\
x_2(k+1) &= x_2(k) + \Delta t x_4(k)\\
x_3(k+1) &= x_3(k) + \Delta t(D(k)x_3(k) + G(k)x_1(k)) + v_1\\
x_4(k+1) &= x_4(k) + \Delta t(D(k)x_4(k) + G(k)x_2(k)) + v_2\\
x_5(k+1) &= x_5(k) + \Delta t v_3,
\end{aligned} \tag{62}$$

Algorithm 2 Square-Root UIF for state estimation

- Initialization:
  $\hat{x}_0 = E(x_0),\ S_{x0} = chol\{E((x_0-\hat{x}_0)(x_0-\hat{x}_0)^T)\},\ S_v = \sqrt{R_v}$ and $S_n = \sqrt{R_n}$.
- For $k = 1, \cdots, \infty$:
  1) Generate sigma points for prediction:

$$\hat{x}_{k-1}^{a_v} = \begin{bmatrix} \hat{x}_{k-1} \\ \bar{v} \end{bmatrix}, \quad S_{k-1}^{a_v} = \begin{bmatrix} S_{x_{k-1}} & 0 \\ 0 & S_v \end{bmatrix} \tag{46}$$

$$\mathscr{X}_{k-1}^{a_v} = \begin{bmatrix} \hat{x}_{k-1}^{a_v} & \hat{x}_{k-1}^{a_v} + \gamma S_{k-1}^{a_v} & \hat{x}_{k-1}^{a_v} - \gamma S_{k-1}^{a_v} \end{bmatrix} \tag{47}$$

  2) Prediction equations:

$$\mathscr{X}_{k|k-1}^{x} = f(\mathscr{X}_{k-1}^{x}, \mathscr{X}_{k-1}^{v}, u_{k-1}) \tag{48}$$

$$\hat{x}_k^- = \sum_{i=0}^{2L} w_i^{(m)} \mathscr{X}_{i,k|k-1}^{x} \tag{49}$$

$$S_{x_k}^- = qr\left\{ \sqrt{w_1^{(c)}} \left(\mathscr{X}_{1:2L,k|k-1}^{x} - \hat{x}_k^- \right) \right\} \tag{50}$$

$$C = sign\{w_0^{(c)}\} \sqrt{w_0^{(c)}} \left(\mathscr{X}_0^{x} - \hat{x}_k^- \right) \tag{51}$$

$$S_{x_k}^- = cholupdate\left\{ S_{x_k}^-, C, +1 \right\} \tag{52}$$

$$\hat{y}_k^- = \left(S_{x_k}^-\right)^T \backslash \left(S_{x_k}^- \backslash \hat{x}_k^- \right) \tag{53}$$

$$S_{y_k}^- = qr\left\{ S_{x_k}^- \backslash I \right\} \tag{54}$$

  3) Generate sigma points for measurement update:

$$\mathscr{X}_{k|k-1} = \begin{bmatrix} \hat{x}_k^- & \hat{x}_k^- + \gamma S_{x_k}^- & \hat{x}_k^- - \gamma S_{x_k}^- \end{bmatrix} \tag{55}$$

  4) Measurement update equations:

$$\mathscr{Z}_{k|k-1} = h\left(\mathscr{X}_{k|k-1} \right) \tag{56}$$

$$\hat{z}_k^- = \sum_{i=0}^{2L} w_i^{(m)} \mathscr{Z}_{i,k|k-1} \tag{57}$$

$$P_{x_k z_k} = \sum_{i=0}^{2L} w_i^{(c)} [\mathscr{X}_{i,k|k-1} - \hat{x}_k^-][\mathscr{Z}_{i,k|k-1} - z_k^-]^T \tag{58}$$

$$U = \left(S_{x_k}^-\right)^T \backslash \left(S_{x_k}^- \backslash P_{x_k z_k} \right) / S_n \tag{59}$$

$$y_k = \hat{y}_k^- + U / S_n^T (z_k - \hat{z}_k^- + P_{x_k z_k}^T \hat{y}_k^-) \tag{60}$$

$$S_{y_k} = cholupdate\{ S_{y_k}^-, U, +1 \} \tag{61}$$

where v_1, v_2 and v_3 are Gaussian process noises, $D(k)$ is the drag-related force, $G(k)$ is the gravity-related force, and $\Delta t = 0.1s$ is the sampling time. The force terms are given by

$$\begin{aligned} D(k) &= \beta(k) V(k) exp\left\{ \tfrac{R_0 - R(k)}{H_0} \right\} \\ G(k) &= -\tfrac{Gm_0}{R^3(k)}, \end{aligned} \tag{63}$$

where $\beta(k) = \beta_0\, exp\{x_5(k)\}$, $R(k) = \sqrt{x_1^2(k) + x_2^2(k)}$ is the distance between the vehicle and the earth center, and $V(k) = \sqrt{x_3^2(k) + x_4^2(k)}$ is the vehicle's speed. The constants in (63) are defined as: $\beta_0 = -0.59783, H_0 = 13.406, Gm_0 = 3.9860 \times 10^5, R_0 = 6374$. The discrete process noise covariance in our simulation is defined by

$$R_v = diag(2.4064 \times 10^{-5}, 2.4064 \times 10^{-5}, 10^{-6}), \tag{64}$$

where *diag* means the diagonal matrix. The vehicle is tracked by two radars which are located at (x_s, y_s), where $s = 1, 2$, and the measurements model is

$$\begin{aligned} r_s(k) &= \sqrt{(x_1(k) - x_s)^2 + (x_2(k) - y_s)^2} + e_{r,s} \\ \theta_s(k) &= tan^{-1}\left(\tfrac{x_2(k) - y_s}{x_1(k) - x_s} \right) + e_{\theta,s}, \end{aligned} \tag{65}$$

where $[e_{r,s}, e_{\theta,s}]^T \sim \mathscr{N}(0, R_{n,s})$ is the measurement noise. In the simulation, the radars are located at $(x_1, y_1) = (6474, 0)$ and $(x_2, y_2) = (6475, -30)$, and their measurement noise variances are

$$\begin{aligned} R_{n,1} &= diag((1 \times 10^{-3})^2, (1.7 \times 10^{-4})^2) \\ R_{n,2} &= diag((2 \times 10^{-3})^2, (1.7 \times 10^{-4})^2). \end{aligned} \tag{66}$$

The initial true state and the covariance of the vehicle are given by

$$\begin{aligned} x_0 &= [6500.4, 349.14, -1.8093, -6.7967, 0.6932]^T \\ P_0 &= diag(10^{-6}, 10^{-6}, 10^{-6}, 10^{-6}, 0), \end{aligned} \tag{67}$$

and the prior state and the covariance are given by

$$\begin{aligned} \hat{x}_0 &= [6500.4, 349.14, -1.8093, -6.7967, 0]^T \\ \hat{P}_0 &= diag(10^{-6}, 10^{-6}, 10^{-6}, 10^{-6}, 1), \end{aligned} \tag{68}$$

which are the same as those used in [11].

The time step Δt in (62) is set to $0.1s$, and measurements from both radars are received during each step, such that the observation frequency of both radars is $10Hz$.

The results of the simulation are derived from 100 Monte Carlo simulations, and summarized in Table I, where UIFa, CDIFa, SRUIFa and SRCDIFa consider only the measurement from the first radar, and UIFb, CDIFb, SRUIFb and SRCDIFb consider measurements from both radars. The results indicate that by fusing two sensors, the CDIF and UIF can achieve much more accurate results, i.e., the mean and the standard deviation of the root mean square error (RMSE) of the position decrease, whereas the additional computational cost for fusion is very low, e.g., only 0.36% for SRCDIF [2]. Furthermore, the SRCDIF has the lowest additional computational cost for sensor fusion in this simulation, although all filters have almost identical RMSE over time (at least to the fourth decimal place).

Because the UIFs and CDIFs have identical estimation results, we only show the state estimation errors from the SRCDIFa and SRCDIFb in Fig. 1. It can been seen that the convergence rate of the SRCDIFb is faster than that of the SRCDIFa, and the SRCDIFb has a better estimation accuracy in most cases, except Fig. 1e where the aerodynamic force parameter $x5$ of the SRCDIFb has a larger RMSE error when $t > 50s$.

[2]The additional computational cost is computed as follows: $(T_b - T_a)/T_a$ where T_a and T_b are average run times using one sensor and two sensors respectively. For instance, the additional computational cost for SRCDIF is derived by $0.36\% = (4.5743 - 3.5022)/3.5022$ from the Table I.

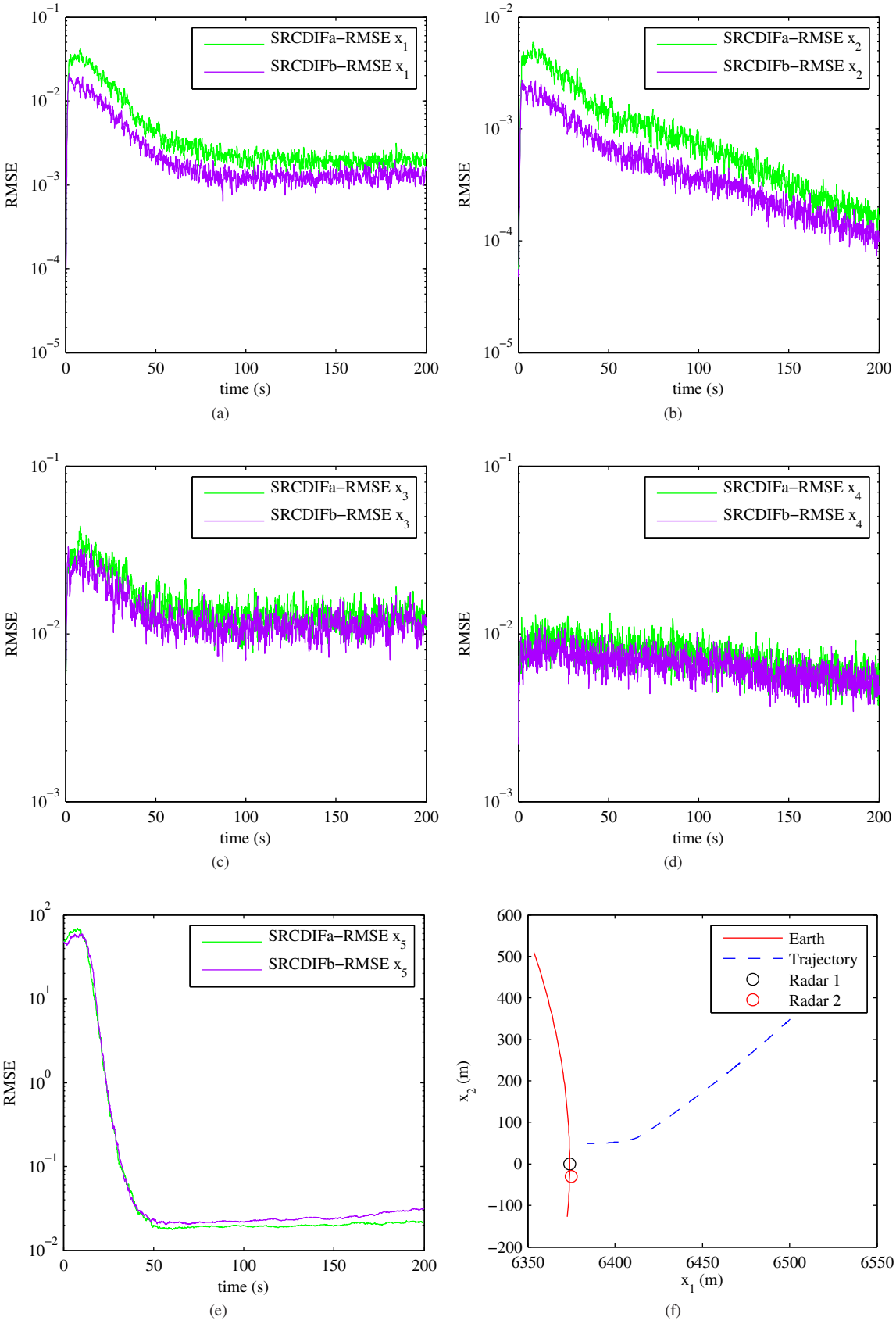


Fig. 1. (a)-(e): the RMSE error of x_1, x_2, x_3, x_4 and x_5 against the time. (f): the trajectory of the vehicle and the positions of radars.

TABLE I
MEANS (E) AND STANDARD DEVIATIONS (STD) OF RMSE VALUES OF THE POSITION AND AVERAGE RUN TIME IN 100 MONTE CARLO RUNS OF THE SPACE-TRACKING PROBLEM

Number of Sensors	Method	E[RMSE]	STD[RMSE]	Average run time(s)
One	UIFa	0.0083	0.0007	4.0632
	CDIFa	0.0083	0.0007	3.5140
	SRUIFa	0.0083	0.0007	4.0667
	SRCDIFa	0.0083	0.0007	**3.5022**
Two	UIFb	0.0060	0.0005	5.3660
	CDIFb	0.0060	0.0005	4.6454
	SRUIFb	0.0060	0.0005	5.3364
	SRCDIFb	0.0060	0.0005	**4.5743**

B. Bearing-only tracking

In this section, we consider a nonlinear bearing-only tracking (BOT) problem using the UIF, CDIF, SRUIF and SRCDIF and compare their performances. The bearing-only tracking problem has become an important benchmark for different probability inference methods. By solving a BOT problem on a moving sensor platform, Bar-Shalom et al. [13] analyzed the performance of the Taylor linearization in the EKF, Lin *et al.* [14] have compared the performance of the EKF, pseudo-measurement filter and particle filter, and Sadhu *et al.* [15] proposed a new track-loss criterion for the comparison between the EKF and the square-root UKF. In addition, Hartikainen and Särkkä [16] have developed a toolbox which includes the comparison between the UKF, the EKF, and their smoothers by solving the BOT problem with static sensors.

Here we use the same system model as in [16]. A moving target object is tracked by two static angular sensors. The discrete time update of the dynamic object on time step k is

$$\mathbf{x}_k = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_{k-1} + \mathbf{v}_{k-1} \tag{69}$$

where the system state is $\mathbf{x}_k = (x_k, y_k, \dot{x}_k, \dot{y}_k)^T$, which includes the target position (x_k, y_k) and velocity $(\dot{x}_k, \dot{y}_k)$. Δt is the time interval between time step k and $k-1$, which is set to $\Delta t = 0.01$ in the simulation. $\mathbf{v}_{k-1}$ is Gaussian noise with zero mean and the covariance is

$$\mathbf{R}_v = \begin{bmatrix} \frac{1}{3}\Delta t^3 & 0 & \frac{1}{2}\Delta t^2 & 0 \\ 0 & \frac{1}{3}\Delta t^3 & 0 & \frac{1}{2}\Delta t^2 \\ \frac{1}{2}\Delta t^2 & 0 & \Delta t & 0 \\ 0 & \frac{1}{2}\Delta t^2 & 0 & \Delta t \end{bmatrix} \beta \tag{70}$$

where β is the spectral density of the noise [16] and set to $\beta = 0.1$ in our experiment. The target is tracked by sensors located at (x_s, y_s), where $s = 1, 2$ in the case of two sensors. The measurement model of the s_{th} sensor is defined as

$$\theta_s = tan^{-1}\left(\frac{y_k - y_s}{x_k - x_s}\right) + e_{\theta,s} \tag{71}$$

where $e_{\theta,s} \sim \mathcal{N}(0, R_{n,s})$ is the measurement noise of the s_{th} sensor. The sensors are located at $(x_1, y_1) = (-1, -2)$ and $(x_2, y_2) = (1, 1)$, and their measurement noise variances are $R_{n,1} = R_{n,2} = 0.05^2$. The initial prior state $\hat{\mathbf{x}}_0$ and the covariance $\hat{\mathbf{P}}_0$ are given by:

$$\hat{\mathbf{x}}_0 = [0,\ 0,\ 1,\ 0]^T \tag{72}$$

$$\hat{\mathbf{P}}_0 = diag(0.1, 0.1, 10, 10). \tag{73}$$

To achieve a curved trajectory the target has a randomized acceleration in our simulation [16]. The estimated results from different filters are summarized in Table II. It can be seen that the UIF and SRUIF have equal accuracy, as do the CDIF and the SRCDIF. Here we only show the comparison between the SRUIF and SRCDIF in Fig. 2. When only one sensor is available, the filters are hardly able to track the target and have very large RMSE errors as can be seen in Fig. 2a. Although the CDIFs still run faster than the UIFs in this simulation, the CDIFs have larger errors and covariances at the beginning of the trajectory. The filters achieve better results by fusing one more sensor which is shown in Fig. 2b.

VI. CONCLUSION

In this paper, a new central difference information filter (CDIF) algorithm for multiple sensor fusion and target tracking was presented. It is analogous to the UIF, but uses Stirling's interpolation instead of the unscented transform. Therefore, the CDIF only depends on one parameter (interval size) in contrast to three parameters which are required in the unscented transform. This makes the CDIF simpler, faster and easier tune than the UIF. Furthermore, the square-root forms of the CDIF and UIF are presented and discussed. They have equal complexity as the original CDIF and UIF, i.e., $\mathcal{O}(L^3)$, however, the SRCDIF and SRUIF have better numerical properties and guarantee positive semi-definitives of the state covariance. In the presented experiments, the CDIFs are faster than the UIFs, but the square-root forms do not improve the computational cost as we expected (the square-root UKF is 20% faster than the UKF as shown in [9]). In the future, we plan to investigate their performances with different sensor network architectures [1], and further improve the estimation accuracies, e.g., by combining the proposed filters with the adaptive consensus algorithm [2], [17].

ACKNOWLEDGMENT

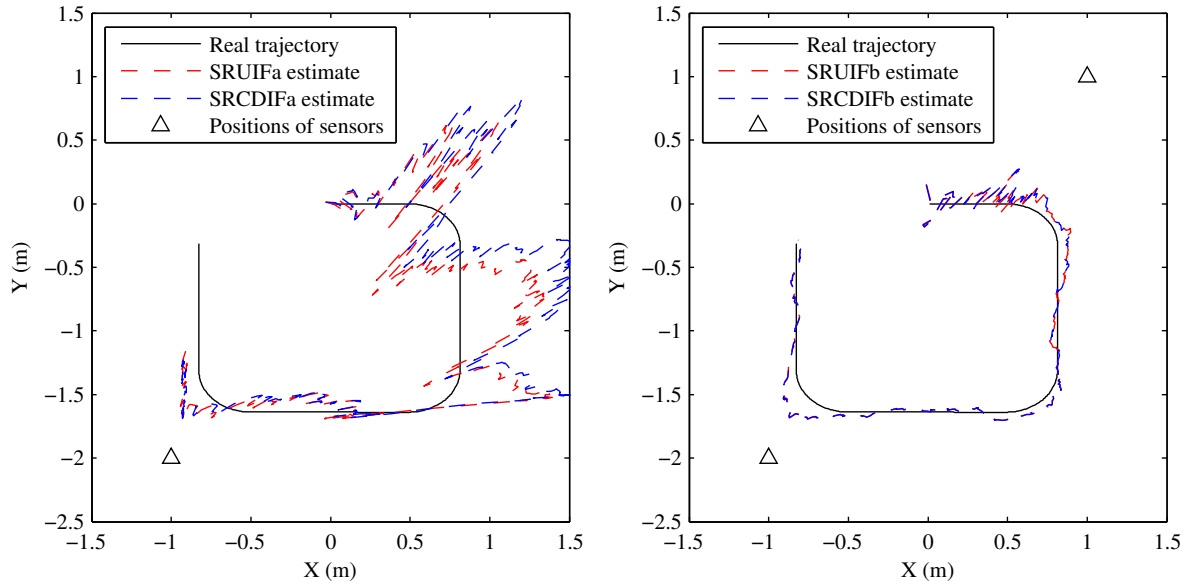
The authors would like to thank Jeremie Papon for his valuable comments.

REFERENCES

[1] D.-J. Lee, "Unscented information filtering for distributed estimation and multiple sensor fusion," in *AIAA Guidance, Navigation, and Control Conference*, 2008.

TABLE II
MEANS (E) AND STANDARD DEVIATIONS (STD) OF RMSE VALUES OF THE POSITION AND AVERAGE RUN TIME IN 100 MONTE CARLO RUNS OF THE BEARING-ONLY TRACKING

Number of Sensors	Method	E[RMSE]	STD[RMSE]	Average run time(*s*)
One	UIFa	0.6435	0.1364	0.5169
	CDIFa	0.6435	0.1638	**0.3512**
	SRUIFa	0.6647	0.1364	0.5042
	SRCDIFa	0.6647	0.1638	0.3596
Two	UIFb	0.1127	0.0277	0.8240
	CDIFb	0.1147	0.0294	0.5760
	SRUIFb	0.1127	0.0277	0.7952
	SRCDIFb	0.1147	0.0294	**0.5736**



(a) (b)

Fig. 2. Comparison between the ground truth and the estimated trajectory for bearing-only tracking. (a): the comparison results when only one sensor is available. (b): the comparison results by fusing two sensors.

[2] L. Wang, Q. Zhang, H. Zhu, and L. Shen, "Adaptive consensus fusion estimation for msn with communication delays and switching network topologies," in *Decision and Control (CDC), 2010 49th IEEE Conference on*, dec. 2010, pp. 2087 –2092.

[3] D.-J. Lee, "Nonlinear estimation and multiple sensor fusion using unscented information filtering," *Signal Processing Letters, IEEE*, vol. 15, pp. 861 –864, 2008.

[4] R. Van der Merwe, "Sigma-point kalman filters for probabilistic inference in dynamic state-space models," Ph.D. dissertation, OGI School of Science & Engineering, Oregon Health & Science University,, 2004.

[5] Y. Kim, J. Lee, H. Do, B. Kim, T. Tanikawa, K. Ohba, G. Lee, and S. Yun, "Unscented information filtering method for reducing multiple sensor registration error," in *Multisensor Fusion and Integration for Intelligent Systems, IEEE International Conference on*, aug. 2008, pp. 326 –331.

[6] G. Liu, F. Wörgötter, and I. Marklić, "Nonlinear estimation using central difference information filter," in *IEEE Workshop on Statistical Signal Processing*, 2011, pp. 593–596.

[7] L. Wang, N. Wang, and H. Zhu, "Consensus based distributed unscented information filtering for air mobile sensor networks," in *Informatics in Control, Automation and Robotics (CAR), 2010 2nd International Asia Conference on*, vol. 2, mar. 2010, pp. 492 –495.

[8] J. Zhu, N. Zheng, Z. Yuan, Q. Zhang, X. Zhang, and Y. He, "A slam algorithm based on the central difference kalman filter," in *Intelligent Vehicles Symposium, 2009 IEEE*, june 2009, pp. 123 –128.

[9] R. Van der Merwe and E. Wan, "The square-root unscented kalman filter for state and parameter-estimation," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, vol. 6, 2001, pp. 3461 –3464 vol.6.

[10] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, September 2005.

[11] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401 – 422, mar 2004.

[12] S. Särkkä, "Unscented rauch–tung–striebel smoother," *Automatic Control, IEEE Transactions on*, vol. 53, no. 3, pp. 845 –849, april 2008.

[13] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. Wiley Interscience, 2001.

[14] X. Lin, T. Kirubarajan, Y. Bar-Shalom, and S. Maskell, "Comparison of ekf, pseudomeasurement, and particle filters for a bearing-only target tracking problem," in *in Proc. SPIE Conference of Signal and Data Processing of Small Targets*, 2002.

[15] S. Sadhu, M. Srinivasan, S. Mondal, and T. Ghoshal, "Bearing only tracking using square root sigma point kalman filter," in *India Annual Conference, 2004. Proceedings of the IEEE INDICON 2004. First*, dec. 2004, pp. 66 – 69.

[16] J. Hartikainen and S. Särkkä, "Optimal filtering with kalman filters and smoothers a manual for matlab toolbox ekf/ukf," Department of Biomedical Engineering and Computational Science, Helsinki University of Technology, Finland, feb. 2008.

[17] D. Casbeer and R. Beard, "Distributed information filtering using consensus filters," in *American Control Conference, 2009. ACC '09.*, june 2009, pp. 1882 –1887.