18th International Conference on Information Fusion
Washington, DC - July 6-9, 2015

Comparison of Angle-only Filtering Algorithms in 3D Using EKF, UKF, PF, PFF, and Ensemble KF

Syamantak Datta Gupta[1], Jun Ye Yu[2], Mahendra Mallick[2], Mark Coates[4], Mark Morelande[5]

[1]McGill University, Montreal, Quebec, Canada, syamantak.dattagupta@mcgill.ca
[2]McGill University, Montreal, Quebec, Canada, jun.y.yu@mail.mcgill.ca
[3]Independent Consultant, Anacortes, WA., USA, mahendra.mallick@gmail.com
[4]McGill University, Montreal, Quebec, Canada, mark.coates@mcgill.ca
[5]National Australia Bank, Melbourne, VIC, Australia, m.morelande@gmail.com

Abstract— **In our previous work, we compared the performance of the extended Kalman filter (EKF), unscented Kalman filter (UKF), and particle filter (PF) for the angle-only filtering (AOF) problem in 3D using Cartesian coordinates and modified spherical coordinates (MSC) for the relative state vector. We found that the UKF-MSC and EKF-MSC had the best performance in accuracy, the UKF-MSC being slightly better than the EKF-MSC. The PF didn't perform well compared with the EKF and UKF and had a higher computational cost. In this work, we compare the performance of the particle flow filter (PFF) with the other filters for the AOF problem. In addition, we also analyze the performance of two versions of the ensemble Kalman filter (EnKF) in this comparative study. We present numerical results from Monte Carlo simulations to analyze the state estimation accuracy and computational cost of these filters.**

Keywords: Angle-only filtering (AOF) in 3D, Modified spherical coordinates (MSC), Extended Kalman filter (EKF), Unscented Kalman filter (UKF), Particle filter (PF), Particle flow filter (PFF), and Ensemble Kalman filter (EnKF).

I. Introduction

The angle-only filtering (AOF) problem in 3D using bearing and elevation angles occurs in a number of important real-world problems. These problems include passive ranging using an infrared search and track (IRST) sensor, passive sonar, passive radar in the presence of jamming [6], satellite-to-satellite passive tracking, ballistic object (a missile or satellite) tracking by optical telescopes [10], and ground target tracking by small unmanned aerial vehicles by video cameras [34]. Significant research has been performed for the bearings-only filtering (BOF) problem in 2D, see [6], [35] and the references therein. However, the number of papers on the AOF problem in 3D is relatively small [1], [22], [28], [31]— [33], [34], [36], [38], [39].

Three types of coordinate systems, Cartesian coordinates, modified spherical coordinates (MSC), and log spherical coordinates (LSC) are used to represent the relative state vector [31]–[33], [38] in the AOF problem. The MSC and LSC are the 3D analogue of the MPC [35] and LPC [8], respectively, for the BOF problem. The components of the Cartesian coordinates are 3D position and velocity. Stallard [38] first proposed the modified spherical coordinates (MSC) for the AOF problem. The elements of MSC are elevation, elevation-rate, bearing, bearing-rate times cosine of elevation, range-rate divided by range, and the inverse of range [31]–[33]. The LPC were proposed in [31], whose first five elements were the same as the first five of the MSC. The sixth element of the LSC is the logarithm of range. Numerical results in [31] show that the EKF-MSC and EKF-LSC have nearly the same estimation accuracy. Therefore, the MSC only were used in [32], [33] and in this paper we only study the MSC.

The discrete-time EKF using Cartesian coordinates for the relative state vector in the near-constant velocity model [4] in 3D and nonlinear measurement model for bearing and elevation does not face the collapse of the covariance matrix as in the BOF. However, as in MPC, the dynamic model using MSC is nonlinear and complex. Since the bearing and elevation are components of the MSC, the measurement model is linear and the measurement update step in MSC is straightforward [4].

The particle flow filter (PFF) developed by Daum and Huang [10], [11], [15] has received a great deal attention in the Fusion community because it has been claimed to have superior accuracy and computational speed. This has led to publications by other researchers [5], [9], [16], [29], [30]. In many cases, detailed steps for implementing the PFF algorithm are missing; the paper [16] provides algorithm details for the benefit of a graduate student or researcher.

Daum and Huang studied the AOF problem of a long-range ballistic missiles using angle-only measurements in [10] and used the MSC for the relative state vector. They compared the performance of the EKF-MSC and the particle flow filter (PFF) using MSC (PFF-MSC) using a time interval of 400 seconds. They used 500 particles in the PFF-MSC and used 100 Monte Carlo simulations. They studied various geometries by varying the angle between the velocity of the missile and the line-of-sight (LOS) of the sensor and presented the median error for missile position. Details on the missile dynamic model, sensor trajectory, and the measurement accuracy were not presented in [10]. It was claimed that for certain difficult geometries, the PFF-MSC was more accurate than the EKF-MSC by an order of magnitude [10], [15]. A close examination of the plots in Figs. 2–6, however, shows that the median position error

for the EKF-MSC and PFF-MSC were nearly the same after 200 - 300 seconds, and the EKF-MSC had a lower median position error than the PFF-MSC. Only in two geometries (25 deg and 10 deg), the EKF-MSC had a higher median position error than the PFF-MSC around 100 s and 150 s, respectively. For these two geometries, the EKF-MSC had lower median position error after 200 s and 300 s, respectively.

Exact SDEs for MSC and LSC were derived in [31], from the near-constant velocity model in 3D for the relative Cartesian state vector. Then EKFs were implemented for MSC and LSC by numerically integrating nonlinear differential equations for the predicted state estimate and covariance matrix jointly. This technique is an example of continuous-discrete filtering (CDF).

Although we use a dynamic model that is identical to that analyzed in [1], [38], in their EKF implementations the process noises are not properly accounted for. Consequently, the approximate method used in [1], [38] to calculate the predicted covariance is not valid over time intervals where the relative geometry between the target and ownship changes substantially. In this paper, the predicted state estimates and covariances are evaluated using the second approach for the exact discrete-time dynamic model in MSC.

Two classes of filtering algorithms using Cartesian coordinates and MSC for the relative state vector [32], [33] are used in this paper. The dynamic and measurement models are discrete in each class. For algorithms using Cartesian coordinates, the near-constant velocity model in 3D is linear and the measurement model is nonlinear. In MSC, the dynamic model is nonlinear and the measurement model is linear. An EKF [4], [19], an UKF [26], [27], [35] and a bootstrap particle filter (BPF) [3], [21], [35] are used for each class.

To initialize the filters, we employ the algorithms described in [32], [33]. Details of the EKF (Cartesian EKF or CEKF and EKF-MSC) [4], [19] the UKF [26], [27] and the BPF [3], [21], [35] for this 3D angle-only setting can be found in [33] and we do not repeat them in this paper. Furthermore, as the MSC version of an algorithm differs only in the propagation step and in evaluation of the predicted observation, we have not included separate algorithms for the MSC versions.

The rest of the paper is organized as follows. In Section II, we introduce the tracker and sensor coordinate frames and discuss different coordinate systems for the target and ownship. The dynamic models of the target and ownship and the measurement models using the Cartesian coordinates and MSC are provided in Section III. In Section IV, algorithms for the Cartesian PFF (CPFF), Cartesian PFF (CPFF-localized), Cartesian deterministic EnKF (CDEnKF) and Cartesian stochastic EnKF (CSEnKF) are discussed. These algorithms had previously not been used for this problem. Numerical results based on simulations are presented in Section V. Finally our conclusions are summarized in Section VI.

II. Coordinate Systems for Target and Ownship

The origin of the tracker coordinate frame (T frame) is specified by the geodetic longitude λ_0, geodetic latitude ϕ_0, and geodetic height h_0. As indicated in Figure 1 the X, Y, and Z axes of the T frame are directed along the local east, north, and upward directions, respectively. We assume that the ownship's motion is deterministic with known parameters and its position is known. The ownship performs maneuvers so that the target state becomes observable. Since we use standard conventions for coordinate frames and angle variables, the rotational transformation $\mathbf{T}_{\mathrm{T}}^{\mathrm{S}}$ from the T frame to the sensor frame (S frame) is defined differently in our approach compared with that in [38].

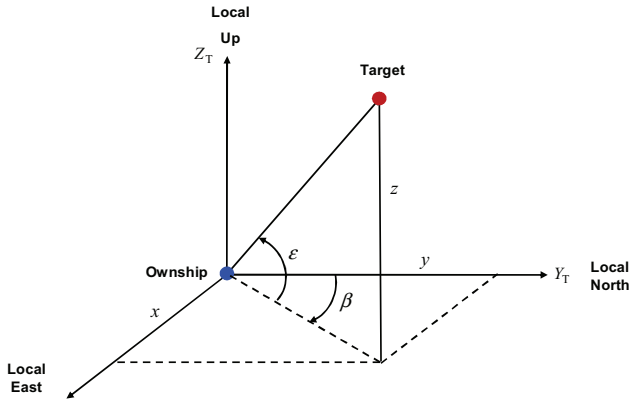

Fig. 1: Definition of tracker coordinate frame (T frame), bearing $\beta \in [0, 2\pi]$ and elevation angle $\epsilon \in [-\pi/2, \pi/2]$.

A. Cartesian Coordinates for Relative State Vector

The state variables of the target and the ownship are defined as follows, in Cartesian co-ordinates:

$$\mathbf{x}^{\mathrm{t}} := \begin{bmatrix} x^{\mathrm{t}} & y^{\mathrm{t}} & z^{\mathrm{t}} & \dot{x}^{\mathrm{t}} & \dot{y}^{\mathrm{t}} & \dot{z}^{\mathrm{t}} \end{bmatrix}', \tag{1}$$

$$\mathbf{x}^{\mathrm{o}} := \begin{bmatrix} x^{\mathrm{o}} & y^{\mathrm{o}} & z^{\mathrm{o}} & \dot{x}^{\mathrm{o}} & \dot{y}^{\mathrm{o}} & \dot{z}^{\mathrm{o}} \end{bmatrix}'. \tag{2}$$

The state vector of the target with respect to the ownship is defined by

$$\mathbf{x} := \mathbf{x}^{\mathrm{t}} - \mathbf{x}^{\mathrm{o}}. \tag{3}$$

Let $\mathbf{x} = [x \quad y \quad z \quad \dot{x} \quad \dot{y} \quad \dot{z}]'$. Let the range vector of the target from the ownship be $\mathbf{r}^{\mathrm{T}}$. Then $\mathbf{r}^{\mathrm{T}}$ is defined by

$$\mathbf{r}^{\mathrm{T}} := \begin{bmatrix} x & y & z \end{bmatrix}' = \begin{bmatrix} x^{\mathrm{t}} - x^{\mathrm{o}} & y^{\mathrm{t}} - y^{\mathrm{o}} & z^{\mathrm{t}} - z^{\mathrm{o}} \end{bmatrix}'. \tag{4}$$

The range is defined by

$$r := \|\mathbf{r}^{\mathrm{T}}\| = \sqrt{x^2 + y^2 + z^2}, \quad r > 0. \tag{5}$$

The range vector can be expressed in terms of range, bearing (β) and elevation (ϵ), as defined in Figure 1, by

$$\mathbf{r}^{\mathrm{T}} = r \begin{bmatrix} \cos\epsilon \sin\beta \\ \cos\epsilon \cos\beta \\ \sin\epsilon \end{bmatrix}, \quad \beta \in [0, 2\pi], \quad \epsilon \in [-\pi/2, \pi/2]. \tag{6}$$

Finally, the ground range is defined as follows:

$$\rho := \sqrt{x^2 + y^2} = r\cos\epsilon, \quad \rho > 0. \tag{7}$$

B. Modified Spherical Coordinates for Relative State Vector

We follow the conventions of Stallard [38], and define ω as a component of the MSC, where

$$\omega(t) := \dot{\beta}(t)\cos\epsilon(t). \tag{8}$$

Let $\zeta(t)$ denote the logarithm of range $r(t)$

$$\zeta(t) := \ln\, r(t). \tag{9}$$

Then

$$r(t) = \exp[\zeta(t)]. \tag{10}$$

Differentiating (9) with respect to time, we get

$$\dot{\zeta}(t) = \dot{r}(t)/r(t). \tag{11}$$

The relative state vector of the target in MSC is defined by [31], [38]

$$\boldsymbol{\xi}(t) := \begin{bmatrix} =\omega(t) & \dot{\epsilon}(t) & \dot{\zeta}(t) & \beta(t) & \epsilon(t) & 1/r(t) \end{bmatrix}'. \tag{12}$$

The components of the MSC defined in [38] and [31] are the same; however, the ordering is different.

III. Dynamic Models

We begin by presenting the discrete-time dynamic models of the target and ownship in the T frame. Next, we present the corresponding dynamic models. The target follows the near-constant velocity model in 3D, while the ownship follows a sequence of constant velocity (CV) and coordinated turn (CT) models in a plane parallel to the XY plane of the T frame.

A. Dynamic Model for State Vector and Relative State Vector in Cartesian Coordinates

At time t_k, let the Cartesian state vector of the target be given by $\mathbf{x}^{\mathrm{t}}_k$. Following [4], [19], the discrete-time dynamic model of the target can then be defined as follows

$$\mathbf{x}^{\mathrm{t}}_k = \mathbf{F}_{k-1}\mathbf{x}^{\mathrm{t}}_{k-1} + \mathbf{w}^{\mathrm{t}}_{k-1}, \tag{13}$$

where $\mathbf{F}_{k-1}$ and $\mathbf{w}^{\mathrm{t}}_{k-1}$ are the state transition matrix [4], [19] and integrated process noise [19], respectively, for the time interval $[t_{k-1}, t_k]$,

$$\mathbf{F}_{k-1} = \mathbf{F}(t_k, t_{k-1}) := \begin{bmatrix} 1 & \Delta_k \\ 0 & 1 \end{bmatrix} \otimes \mathbf{I}_3, \tag{14}$$

$$\mathbf{w}^{\mathrm{t}}_{k-1} := \int_{t_{k-1}}^{t_k} \mathbf{F}(t_k - t)\mathbf{w}^{\mathrm{t}}(t)\, dt, \tag{15}$$

where $\Delta_k := t_k - t_{k-1}$ is the sampling interval and $\otimes$ refers to the Kronecker product. Using the properties of the continuous-time process noise $\mathbf{w}^{\mathrm{t}}(t)$, state transition matrix $\mathbf{F}_{k-1}$ defined in (14), and the definition of $\mathbf{w}^{\mathrm{t}}_{k-1}$ in (15), it can be shown that $\mathbf{w}^{\mathrm{t}}_{k-1}$ is a zero-mean Gaussian white integrated process noise with covariance $\mathbf{Q}_{k-1}$

$$\mathbf{w}^{\mathrm{t}}_{k-1} \sim N(\mathbf{w}^{\mathrm{t}}_{k-1}; \mathbf{0}, \mathbf{Q}_{k-1}), \tag{16}$$

$$\mathbf{Q}_{k-1} = \begin{bmatrix} \Delta_k^3/3 & \Delta_k^2/2 \\ \Delta_k^2/2 & \Delta_k \end{bmatrix} \otimes \mathrm{diag}(q_x, q_y, q_z). \tag{17}$$

Here, q_x, q_y, q_z are the power spectral densities of the process noise [19] along the X, Y, and Z axes respectively.

As the dynamic model of the ownship is deterministic, the process noises are not used to simulate its motion. The CV and CT motions of the ownship are defined as follow:

$$\mathbf{x}^{\mathrm{o}}_k = \mathbf{F}_{k-1}\mathbf{x}^{\mathrm{o}}_{k-1}, \tag{18}$$

$$\mathbf{x}^{\mathrm{o}}_k = \mathbf{F}^{\mathrm{CT}}(\Delta_k, \omega)\mathbf{x}^{\mathrm{o}}_{k-1}, \tag{19}$$

where

$$\mathbf{F}^{\mathrm{CT}}(\Delta, \omega) = \begin{bmatrix} 1 & 0 & 0 & \sin(\omega\Delta)/\omega & -[1-\cos(\omega\Delta)]/\omega & 0 \\ 0 & 1 & 0 & [1-\cos(\omega\Delta)]/\omega & \sin(\omega\Delta)/\omega & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta \\ 0 & 0 & 0 & \cos(\omega\Delta) & -\sin(\omega\Delta) & 0 \\ 0 & 0 & 0 & \sin(\omega\Delta) & \cos(\omega\Delta) & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{20}$$

Using (13) in the definition (3) of the relative state vector and then adding and subtracting $\mathbf{F}_{k-1}\mathbf{x}^{\mathrm{o}}_{k-1}$, we get

$$\mathbf{x}_k = \mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{w}^{\mathrm{t}}_{k-1} - \mathbf{u}_{k-1}, \tag{21}$$

where

$$\mathbf{u}_{k-1} := \mathbf{x}^{\mathrm{o}}_k - \mathbf{F}_{k-1}\mathbf{x}^{\mathrm{o}}_{k-1}. \tag{22}$$

We note that when ω^{o}_k approaches zero, $\mathbf{F}^{\mathrm{CT}}(\Delta_k, \omega^{\mathrm{o}}_k)$ reduces to $\mathbf{F}_{k-1}$ and $\mathbf{u}_{k-1}$ becomes zero.

B. Dynamic Model for Relative State Vector in Modified Spherical Coordinates

Let $\mathbf{f}^{\mathrm{MSC}}_{\mathrm{C}} : \mathbb{R}^6 \to \mathbb{R}^6$ denote the transformation from relative Cartesian coordinates to MSC. Similarly, let $\mathbf{f}^{\mathrm{C}}_{\mathrm{MSC}} : \mathbb{R}^6 \to \mathbb{R}^6$ denote the inverse transformation from MSC to relative Cartesian coordinates. Then,

$$\boldsymbol{\xi}_k = \mathbf{f}^{\mathrm{MSC}}_{\mathrm{C}}(\mathbf{x}_k), \tag{23}$$

$$\mathbf{x}_k = \mathbf{f}^{\mathrm{C}}_{\mathrm{MSC}}(\boldsymbol{\xi}_k). \tag{24}$$

Functional forms of $\mathbf{f}^{\mathrm{MSC}}_{\mathrm{C}}(\mathbf{x}_k)$ and $\mathbf{f}^{\mathrm{C}}_{\mathrm{MSC}}(\boldsymbol{\xi}_k)$ are presented in [33]. Then, using (21) in (23), we get

$$\boldsymbol{\xi}_k = \mathbf{f}^{\mathrm{MSC}}_{\mathrm{C}}(\mathbf{F}_{k-1}\mathbf{x}_{k-1} + \mathbf{w}^{\mathrm{t}}_{k-1} - \mathbf{u}_{k-1}). \tag{25}$$

We have

$$\mathbf{x}_{k-1} = \mathbf{f}^{\mathrm{C}}_{\mathrm{MSC}}(\boldsymbol{\xi}_{k-1}). \tag{26}$$

Substitution of (26) in (25) gives

$$\boldsymbol{\xi}_k = \mathbf{f}^{\mathrm{MSC}}_{\mathrm{C}}(\mathbf{F}_{k-1}\mathbf{f}^{\mathrm{C}}_{\mathrm{MSC}}(\boldsymbol{\xi}_{k-1}) + \mathbf{w}^{\mathrm{t}}_{k-1} - \mathbf{u}_{k-1}). \tag{27}$$

Formally, we can write (27) as

$$\boldsymbol{\xi}_k = \mathbf{b}(\boldsymbol{\xi}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}^{\mathrm{t}}_{k-1}), \tag{28}$$

where $\mathbf{b}$ is a nonlinear function of $\boldsymbol{\xi}_{k-1}$, $\mathbf{u}_{k-1}$, and $\mathbf{w}^{\mathrm{t}}_{k-1}$. We note that a closed form analytic expression for the nonlinear function $\mathbf{b}$ is difficult to obtain. However, it is straightforward to calculate the predicted state estimate $\hat{\boldsymbol{\xi}}_{k|k-1}$ approximately given $\hat{\boldsymbol{\xi}}_{k-1|k-1}$, using nested functions in (27). It can be

seen from (28) that the process noise $\mathbf{w}_{k-1}^{\mathrm{t}}$ is nonlinearly transformed in the MSC dynamic model. This makes the EKF and UKF slightly more complicated compared to the usual dynamic models in which the process noise is additive.

C. Measurement Models

The passive sensor collects bearing and elevation measurements $\{\mathbf{z}_k\}$ at discrete times $\{t_k\}$. The measurement model for the bearing and elevation angles using the relative Cartesian state vector $\mathbf{x}_k$ is

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{n}_k, \tag{29}$$

$$\text{where } \mathbf{h}(\mathbf{x}_k) := \begin{bmatrix} \beta_k \\ \epsilon_k \end{bmatrix} = \begin{bmatrix} \tan^{-1}(x_k, y_k) \\ \tan^{-1}(z_k, \rho_k) \end{bmatrix}, \tag{30}$$

where the ground range ρ_k is defined in (7) and $\mathbf{n}_k$ is a zero-mean white Gaussian measurement noise with covariance $\mathbf{R}$

$$\mathbf{n}_k \sim \mathcal{N}(\mathbf{n}_k; \mathbf{0}, \mathbf{R}), \quad \mathbf{R} := \mathrm{diag}(\sigma_\beta^2, \sigma_\epsilon^2). \tag{31}$$

The measurement model for the bearing and elevation angles β_k, ϵ_k using MSC is

$$\mathbf{z}_k = \mathbf{H}\boldsymbol{\xi}_k + \mathbf{n}_k, \tag{32}$$

$$\mathbf{H} := \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{33}$$

IV. Nonlinear Filtering using Cartesian and Modified Spherical Coordinates

The dynamic model (near-constant velocity model in 3D) for Cartesian co-ordinates is linear and the measurement model for bearing and elevation is nonlinear for this case. On the other hand, the dynamic model using MSC is nonlinear as in (27) or (28). In addition, the process noise is not additive. Since, the bearing and elevation are elements of the MSC, the measurement model (32) is linear in this case.

Detailed descriptions of algorithms used for the Cartesian and Spherical (MSC) versions of the UKF, EKF and BPF can be found in [32] and [33]. The BPF algorithm was slightly modified in this study by introducing a resampling step when the effective sample size fell below a certain threshold, as opposed to resampling at every step. In this section, we present the algorithms used for the particle flow filter (PFF) and the ensemble Kalman filter (EnKF) .

A. Particle Flow Filter

In the particle flow filter (PFF), [12], [14], [13], a set of particles are migrated smoothly from the previous to the next time instant using a particle flow derived through a differential equation. The unnormalized conditional probability density can be expressed as a product of two probability density functions as follows:

$$\tilde{p}(\mathbf{x}_k|\mathbf{z}_{1:k}) := p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1}). \tag{34}$$

We denote $p(\mathbf{z}_k|\mathbf{x}_k)$, and $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ by $\psi(\mathbf{x}_k)$ and $g(\mathbf{x}_k)$ respectively. Daum and Huang introduce a function $\phi(\mathbf{x}, \lambda)$ as follows:

$$\phi(\mathbf{x}, \lambda) := \log g(\mathbf{x}_k) + \lambda \log \psi(\mathbf{x}_k). \tag{35}$$

with $\lambda \in [0, 1]$. At each time step, the particle flow is created by varying λ from 0 to 1. The objective is to evaluate $\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\lambda}$ for each particle and migrate them to the next time instant accordingly. In [13], Daum and Huang generalized and improved the filter using the Fokker-Planck equation with zero process noise. Denote $\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\lambda}$ by $f(\mathbf{x}, \lambda)$. Under the condition that the function ϕ is smooth and non-zero, the following relation between the function ϕ and f is derived:

$$\frac{\partial \log \phi}{\partial \mathbf{x}} f(\mathbf{x}, \lambda) + \log(\psi) = -\mathrm{Tr}\left(\frac{\partial f}{\partial \mathbf{x}}\right). \tag{36}$$

The exact flow of the probability density function can be evaluated by solving the above differential equation. For a nonlinear measurement model with Gaussian noise, a closed form solution for $\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\lambda}$ can be obtained [16], using a first order Taylor series approximation of the measurement function, and it is given by

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}\lambda} = \mathbf{A}(\lambda)\mathbf{x} + \mathbf{b}(\lambda), \tag{37}$$

where

$$\mathbf{A} = -\frac{1}{2}\mathbf{P}\mathbf{H}_k^\top(\lambda\mathbf{H}_k\mathbf{P}\mathbf{H}_k^\top + \mathbf{R})^{-1}\mathbf{H}_k, \tag{38}$$

$$\mathbf{b} = (\mathbf{I}+2\lambda\mathbf{A})[(\mathbf{I}+\lambda\mathbf{A})\mathbf{P}\mathbf{H}_k^\top\mathbf{R}^{-1}(\mathbf{z}_k-\mathbf{e}_k)+\mathbf{A}+\overline{\mathbf{x}}_k]. \tag{39}$$

Here, $\overline{\mathbf{x}}_k$ denotes the predicted value of $\mathbf{x}_k$, $\mathbf{P}$ is the predicted covariance matrix, $\mathbf{H}_k := \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}}|_{\mathbf{x}=\mathbf{x}_{k|k-1}^i}$, and $\mathbf{e}_k = h(\mathbf{x}_{k|k-1}) - \mathbf{H}_k(\mathbf{x}_{k|k-1})$ denotes the error induced in approximating $h(\mathbf{x}_k)$ by $\mathbf{H}_k(\mathbf{x}_k)$. The predicted value of the covariance $\mathbf{P}$, can be evaluated by an unscented or extended Kalman filter running in parallel to the particle flow filter to improve performance. Pseudocode used for the particle flow paper can be found in Algorithm 1. A slight modification of the original PFF algorithm by Daum and Huang was suggested in [16], where linearization of the measurement function is performed at each particle location, instead of only at the mean. In this filter, which we refer to as the PFF (localized), the pseudocode from the second **for** loop onwards was replaced with the pseudocode provided in Algorithm 2.

Algorithms 1 and 2 provide the recursion steps for Cartesian co-ordinates. For the MSC versions of these two filters, the difference is in the propagation step, where one has to use (28) to propagate the particles, and in the evaluation of $\mathbf{A}$ and $\mathbf{b}$. As the measurement model (32) is linear, a Taylor series approximation of $h(\cdot)$ is not required for the MSC case, and the term $\mathbf{e}_k$ in (39) is zero.

B. Ensemble Kalman Filter

The ensemble Kalman filter (EnKF) [17], [18], [20], [37], [41] is an extension of the Kalman filter for non-linear models, where instead of a single state estimate the filter maintains a statistical sample of state estimates, termed as the ensemble. This filter executes the task of estimation by minimizing the error covariance, where error statistics are modeled using the ensemble of predicted states. At the forecast step, using the ensemble of forecast states, the sample mean and error covariances are evaluated. The forecast ensemble members

Algorithm 1: A recursion of the particle flow filter with Cartesian coordinates (CPFF).

Input: EKF/UKF estimates $\mathbf{m}_{k-1|k-1}$, covariance matrix $\mathbf{P}_{k-1|k-1}$ and the particle cloud $\{\mathbf{x}_{k-1}^i\}$ at time t_{k-1} and the measurement $\mathbf{z}_k$.
Output: updated EKF/UKF mean $\hat{\mathbf{m}}_{k|k}$, covariance matrix $\mathbf{P}_{k|k}$ and cloud $\{\mathbf{x}_k^i\}$ at time t_k.
for $k = 1, \ldots, T$ **do**
  Propagate particles: $\mathbf{x}_k^i = \mathbf{F}(\Delta_k)\mathbf{x}_{k-1}^i - \mathbf{u}_{k-1}$
  Calculate the sample mean $\overline{\mathbf{x}}_k = \frac{1}{N}\sum_{i=1}^{N} x_k^i$
  Apply UKF/EKF Prediction
  $(\mathbf{m}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}) \rightarrow (\mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1})$
  for $j = 1, \ldots, N_\lambda$ **do**
    Set $\lambda = j\Delta\lambda$
    Calculate $\mathbf{H}_{\overline{\mathbf{x}}}$ by linearizing $h(\cdot)$ at $\overline{\mathbf{x}}_k$
    Calculate $\mathbf{A}$ and $\mathbf{b}$ using (38) and (39)
    for *i=1,..., N* **do**
      Evaluate $\frac{\mathrm{d}\mathbf{x}_k^i}{\mathrm{d}\lambda}$ from (37) for each particle
      Migrate particles $\mathbf{x}_k^i = \mathbf{x}_k^i + \Delta\lambda\frac{\mathrm{d}\mathbf{x}_k^i}{\mathrm{d}\lambda}$
    end
    Re-evaluate sample mean $\overline{\mathbf{x}}_k$ using the updated particles $\{\mathbf{x}_k^i\}$.
  end
  Apply UKF/EKF update :
  $(\mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1}) \rightarrow (\mathbf{m}_{k|k}, \mathbf{P}_{k|k})$.
  Estimate $\hat{\mathbf{x}}_k$ from the particles $\{\mathbf{x}_k^i\}$ (sample mean or robust mean)
  Optional: redraw particles $\mathbf{x}_k^i \sim N(\hat{\mathbf{x}}_k, \mathbf{P}_{k|k})$.
end

are combined to evaluate the ensemble covariance of the predicted observations and the cross covariance of the state and observation ensembles. These are used to compute a single Kalman gain matrix, which is subsequently used to update the ensemble using new observation.

Different versions of the EnKF have been proposed, and these can be classified into two categories, stochastic [17], [18], [20] and deterministic [37], [41]. The two classes are identical in the forecast step and differ only in the analysis step. While the stochastic version of the filter computes the observation error covariance by perturbing the observation, the deterministic version avoids this step.

The stochastic version of the EnKF implemented for this paper follows the steps discussed in [20] and is described in the pseudocode given in Algorithm 3. To implement the deterministic version, we used a Taylor series approximation for the matrix square root, as suggested in [37]. The steps are elaborated in Algorithm 4.

For the relative MSC, these algorithms differ only in the propagation step, where (28) is used to obtain the forecast ensemble of state variables, and in evaluating the corresponding forecast value of the observation, which is evaluated by using the linear measurement equation given in (32).

Algorithm 2: A recursion of the particle flow filter-localized with Cartesian coordinates (CPFF-localized).

for $j = 1, \ldots, N_\lambda$ **do**
  Set $\lambda = j\Delta\lambda$
  Calculate $\mathbf{H}_{\overline{\mathbf{x}}}$ by linearizing $h(\cdot)$ at $\mathbf{x}_k$
  Calculate $\mathbf{A}^i$ and $\mathbf{b}^i$ using (38) and (39)
  for *i=1,..., N* **do**
    Evaluate $\frac{\mathrm{d}\mathbf{x}_k^i}{\mathrm{d}\lambda}$ from (37) for each particle
    Migrate particles $\mathbf{x}_k^i = \mathbf{x}_k^i + \Delta\lambda\frac{\mathrm{d}\mathbf{x}_k^i}{\mathrm{d}\lambda}$
  end
  Re-evaluate $\overline{\mathbf{x}}_k$ using the updated particles $\{\mathbf{x}_k^i\}$.
end
Apply UKF/EKF update :
$(\hat{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1}) \rightarrow (\hat{\mathbf{x}}_{k|k}, \mathbf{P}_{k|k})$.
Estimate $\hat{\mathbf{x}}_k$ from the particles $\{\mathbf{x}_k^i\}$ (sample mean or robust mean)
Set $\mathbf{m}_{k|k} = \hat{\mathbf{x}}_k$ Optional: redraw particles
$\mathbf{x}_k^i \sim N(\hat{\mathbf{x}}_k, \mathbf{P}_{k|k})$.

V. Numerical Simulations and Results

The scenario used in our simulation is the same as in [32], [33], and is similar to that used in [7], with changes to make it three dimensional in nature. The initial height z_1^{o} of the ownship is 10.0 km. The target's initial ground range ρ_1, bearing β_1, and height z_1^{t} are shown in Table I. Then the initial elevation ϵ_1 can be calculated. Table I also shows the target's initial speed s_1, course c_1 in the XY-plane and the Z component of velocity $\dot{z}_1^{\mathrm{t}}$. The target's initial position and velocity in Cartesian coordinates can be found from Table I as $(138/\sqrt{2}, 138/\sqrt{2}, 9)$ km and $297/\sqrt{2}(-1, -1, 0)$ m/s, respectively. The motion of the target is governed by the near-constant velocity model. The power spectral densities (q_x, q_y, q_z) of the zero-mean white acceleration process noise along the X, Y, and Z axes of the T frame are $(0.01, 0.01, 0.0001)$ $\mathrm{m^2 s^{-3}}$, respectively.

The ownship moves in a plane parallel to the XY-plane at a fixed height of 10 km, with a motion profile as described in Table II. In Table II, Δt represents the duration of the segment, $\Delta\phi$ is the total angular change during the segment, and ω^{o} is the angular velocity of the ownship during the segment. The ownship trajectory and the true target trajectory from the first Monte Carlo run are shown in Figure 2.

The measurement sampling time interval is 1.0 s. The bearing and elevation measurement error standard deviations used in our simulation were 1, 15 and 35 mrad (0.057, 0.857 and 2 degrees), representing high, medium, and low measurement accuracy, respectively. We used 1000 Monte Carlo simulations to calculate various statistics such as the root mean square (RMS) position and velocity errors.

The filters are initialized as described in [33] with the parameters shown in Table I. PFs are implemented with a sample size of 10,000. The EnKFs and PFFs are run with 500 ensemble members and 500 particles respectively. These numbers were so chosen that increasing them further did not

Algorithm 3: A recursion of the stochastic ensemble Kalman filter with Cartesian coordinates (CSEnKF).

Input: The posterior ensemble $\{\mathbf{x}_{k-1}^i\}$ at time t_{k-1} and the measurement $\mathbf{z}_k$.
Output: updated ensemble $\{\mathbf{x}_k^i\}$ at time t_k.
for $k = 1, \ldots, T$ **do**
  Propagate particles to obtain the forecast state ensemble $\mathbf{x}_k^i = \mathbf{F}(\Delta_k)\mathbf{x}_{k-1}^i - \mathbf{u}_{k-1}$
  Calculate the sample mean of the forecast state ensemble $\overline{\mathbf{x}}_k = \frac{1}{N}\sum_{i=1}^{N} x_k^i$
  for $i = 1, \ldots, N$ **do**
    Compute the predicted observations: $\hat{\mathbf{z}}_k^i = h(\mathbf{x}_k^i)$
    Perturb the observations by adding noise $\hat{\mathbf{z}}_k^i = \hat{\mathbf{z}}_k^i + \mathbf{n}_k^i$ where $\mathbf{n}_k^i \sim N(\mathbf{0}, \mathbf{R})$
  end
  Calculate the sample mean of the forecast observation ensemble $\overline{\mathbf{z}}_k = \frac{1}{N}\sum_{i=1}^{N} \hat{\mathbf{z}}_k^i$
  Evaluate the error matrices $\mathbf{E}_k^x$, $\mathbf{E}_k^z$:
  $\mathbf{E}_k^x = [\mathbf{x}_k^1 - \overline{\mathbf{x}}_k \quad \cdots \quad \mathbf{x}_k^N - \overline{\mathbf{x}}_k]$ and
  $\mathbf{E}_k^z = [\mathbf{z}_k^1 - \overline{\mathbf{z}}_k \quad \cdots \quad \mathbf{z}_k^N - \overline{\mathbf{z}}_k]$
  Evaluate the ensemble covariance terms
  $\mathbf{P}_k^{x,z} = \frac{1}{N-1}\mathbf{E}_k^x(\mathbf{E}_k^z)^\top$
  $\mathbf{P}_k^{z,z} = \frac{1}{N-1}\mathbf{E}_k^z(\mathbf{E}_k^z)^\top$
  Compute the Kalman gain $\mathbf{K} = \mathbf{P}_k^{x,z}(\mathbf{P}_k^{z,z})^{-1}$
  Update ensemble members
  for $i = 1, \ldots, N$ **do**
    $\mathbf{x}_k^i = \mathbf{x}_k^i + \mathbf{K}(\mathbf{z}_k - \hat{\mathbf{z}}_k^i)$
  end
end

Algorithm 4: A recursion of the deterministic ensemble Kalman filter with Cartesian coordinates (CDEnKF).

Input: The posterior ensemble $\{\mathbf{x}_{k-1}^i\}$ at time t_{k-1} and the measurement $\mathbf{z}_k$.
Output: updated ensemble $\{\mathbf{x}_k^i\}$ at time t_k.
for $k = 1, \ldots, T$ **do**
  Propagate particles to obtain the forecast state ensemble $\mathbf{x}_k^i = \mathbf{F}(\Delta_k)\mathbf{x}_{k-1}^i - \mathbf{u}_{k-1}$
  Calculate the sample mean of the forecast state ensemble $\overline{\mathbf{x}}_k = \frac{1}{N}\sum_{i=1}^{N} x_k^i$
  for $i = 1, \ldots, N$ **do**
    Compute the predicted observations: $\hat{\mathbf{z}}_k^i = h(\mathbf{x}_k^i)$
  end
  Calculate the sample mean of the forecast observation ensemble $\overline{\mathbf{z}}_k = \frac{1}{N}\sum_{i=1}^{N} \hat{\mathbf{z}}_k^i$
  Evaluate the error matrices, normalized by $\frac{1}{\sqrt{N-1}}$:
  $\mathbf{E}_k^x = \frac{1}{\sqrt{N-1}}[\mathbf{x}_k^1 - \overline{\mathbf{x}}_k \quad \cdots \quad \mathbf{x}_k^N - \overline{\mathbf{x}}_k]$ and
  $\mathbf{E}_k^z = \frac{1}{\sqrt{N-1}}[\mathbf{z}_k^1 - \overline{\mathbf{z}}_k \quad \cdots \quad \mathbf{z}_k^N - \overline{\mathbf{z}}_k]$
  $\mathbf{S} = \mathbf{E}_k^z(\mathbf{E}_k^z)^\top + \mathbf{R}$
  $\mathbf{U} = \mathbf{I} - (\mathbf{E}_k^z)^\top\mathbf{S}^{-1}\mathbf{E}_k^x$
  Use a first order Taylor series approximation to evaluate the square root of $\mathbf{U}$: $\mathbf{T} = \mathbf{I} - 0.5\mathbf{U}$
  Compute the Kalman gain $\mathbf{K} = \mathbf{E}_k^x(\mathbf{E}_k^z)^\top\mathbf{S}^{-1}$
  Update the mean of the state ensemble $\overline{\mathbf{x}}_k = \overline{\mathbf{E}_k^x} + \mathbf{K}(\mathbf{z} - \overline{\mathbf{z}}_k)$
  Update the state error ensemble $\mathbf{E}_k^x = \mathbf{E}_k^x\mathbf{T}$
  Update the ensemble members:
  for $i = 1, \ldots, N$ **do**
    $\mathbf{x}_k^i = \overline{\mathbf{x}}_k + \sqrt{N-1}\mathbf{E}_k^x$
  end
end

lead to any substantial improvement in tracking performance. The Cartesian versions of the PFF were run alongside an EKF to propagate the covariance matrices while the PFF-MSC was implemented using a UKF-MSC.

Two measures of performance are used to compare the algorithms. The first is the RMS position error averaged from the end of the last observer maneuver to the end of the surveillance period. The second performance measure is the RMS position error at the end of the surveillance period. The results are shown in Tables III and IV. For the filters implemented in MSC, these errors are evaluated using the relative Cartesian coordinates calculated exactly from MSC, following Appendix 1B in [33]. The execution times of the algorithms are given in Table V.

The best performance is achieved by the two implementations of the PFF in Cartesian co-ordinates, followed closely by the MSC versions of the PFF, EKF, UKF and the DEnKF. Especially for low measurement accuracy, the performance improvements of these filters over the others are substantial.

Among the two versions of the ensemble Kalman filters, the deterministic version was found to outperform to a relative extent. For both filters, the performance was better when MSC was used as compared to Cartesian co-ordinates.

TABLE I: Initial parameters of target.

Variable	Value
ρ_1 (km)	138.0
β_1 (deg)	45.0
ϵ_1 (deg)	-0.415
z_1^{t} (km)	9.0
s_1 (m/s)	297.0
c_1 (deg)	-135.0
$\dot{z}_1^{\mathrm{t}}$ (m/s)	0.0

It is interesting to note that the particle flow filter demonstrated a significant improvement over the bootstrap particle filter. The former achieves this performance improvement by using only 500 particles, as compared to 10,000 particles used by the bootstrap filter. The PFF (local) algorithm provided marginal improvements over the original PFF and had a higher cost of computation. Regularization of the covariance matrix following [33], and an introduction of a resampling step improved the performance of the PF. The relative performance of the various filters is very similar when we compare the tracking errors for the velocity components of the state.

The error propagation of the various filters with respect to

TABLE II: Motion profile of ownship for various segments.

Time Interval (s)	Δt (s)	$\Delta\phi$ (rad)	Motion Type	ω° (rad/s)
$[[0, 15]$	15	0	CV	0
$[15, 31]$	16	$-\pi/4$	CT	$-\pi/64$
$[31, 43]$	12	0	CV	0
$[43, 75]$	32	$\pi/2$	CT	$\pi/64$
$[75, 86]$	11	0	CV	0
$[86, 102]$	16	$-\pi/4$	CT	$-\pi/64$
$[102, 210]$	108	0	CV	0

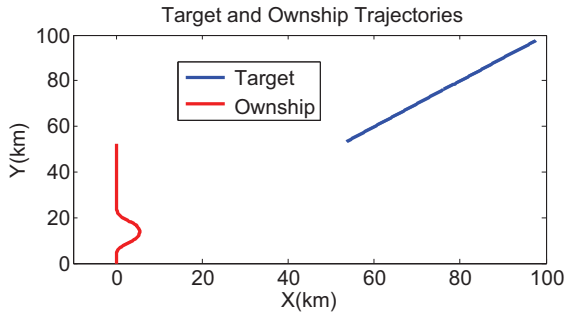

Fig. 2: Target and ownship trajectories.

time is demonstrated in the boxplots provided in Fig. 3.

VI. Conclusions

In this paper, we consider the angle-only filtering problem in 3D using bearing and elevation measurements, and present a comparison of performance of various nonlinear filters using relative Cartesian coordinates and modified spherical coordinates (MSC). Of the algorithms considered here, the EKF, UKF and DEnKF in MSC, and the PFF are found to provide the best performance.

The MSC is the 3D analogue of the well-known modified polar coordinates (MPC) in 2D. In MPC, the bearing, bearing-rate, range-rate divided by range form the observable components of the state vector; while the inverse of range is the unobservable component. Ill-conditioning of the covariance matrix is prevented by decoupling the observable and unobservable components in MPC, which yields better filter performance than that using Cartesian coordinates [23]. For MSC, the first five components are observable and the last component is unobservable. Our results indicate that tracking using MSC is beneficial because of this decoupling, although the benefits are modest in our scenario.

TABLE III: Time-averaged RMS position error in kilometers.

Algorithm	sdv (mrad) 1.0	sdv (mrad) 15.0	sdv (mrad) 35.0
CEKF	0.60	6.30	12.49
CUKF	0.61	5.61	10.64
CBPF	0.68	7.75	15.18
CSEnKF	0.70	7.45	14.23
CDEnKF	0.61	6.25	13.04
CPFF	0.56	4.11	8.55
CPFF(local)	0.55	4.14	8.08
EKF-MSC	0.54	4.46	9.98
UKF-MSC	0.54	4.44	9.99
BPF-MSC	3.63	7.40	12.60
SEnKF-MSC	0.65	5.69	12.78
DEnKF-MSC	0.54	4.48	10.25
PFF-MSC	0.54	4.45	10.07

TABLE IV: Final RMS position error in kilometers.

Algorithm	sdv (mrad) 1.0	sdv (mrad) 15.0	sdv (mrad) 35.0
CEKF	0.41	3.73	5.42
CUKF	0.40	3.51	4.79
CBPF	0.51	6.04	8.39
CSEnKF	0.51	4.45	5.80
CDEnKF	0.42	3.73	5.68
CPFF	0.39	2.09	3.40
CPFF(local)	0.39	2.17	2.78
EKF-MSC	0.38	2.12	2.87
UKF-MSC	0.38	2.09	2.86
BPF-MSC	4.67	5.51	6.94
SEnKF-MSC	0.49	3.09	4.29
DEnKF-MSC	0.38	2.12	2.84
PFF-MSC	0.38	2.05	2.88

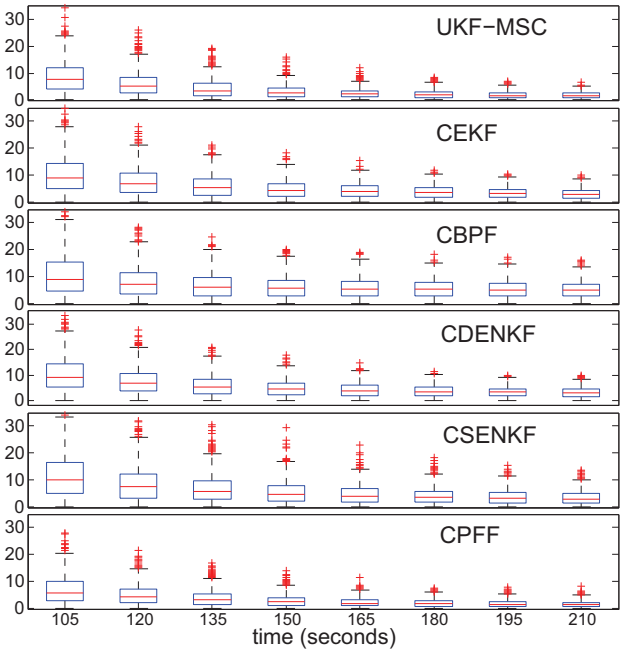

Fig. 3: Boxplot of errors (in kilometer) over time, for the UKF-MSC, CEKF, CBPF, CSENKF, CDENKF and CPFF, at measurement noise sdv = 15 milirad. Boxes indicate 25-75 interquartile range; whiskers extend 1.5 times the range and '+' symbols indicate outliers lying beyond the whiskers.

As extensions of this work, we would like to consider the realistic scenario of a maneuvering target. It would be interesting to calculate the posterior Cramér Rao lower bound (PCRLB) [40], [24] for the filtering problem which represents the best achievable accuracy. This would enable a rigorous comparison of the accuracy of the various filtering algorithms discussed in this paper.

Acknowledgment

The authors thank Sam Blackman for providing the details of the tracking scenario and useful discussions and advice.

TABLE V: CPU times of filtering algorithms for the measurement standard deviation of 15 mrad.

Algorithm	CPU (s)	Relative CPU
CEKF	0.023	1.00
CUKF	0.052	2.26
CSEnKF	0.090	3.91
UKF-MSC	0.136	5.91
SEnKF-MSC	0.201	8.74
CBPF	0.503	21.87
CDEnKF	0.977	42.48
DEnKF-MSC	1.047	45.52
EKF-MSC	1.142	49.65
CPFF	1.257	54.65
BPF-MSC	1.270	55.21
PFF-MSC	1.335	58.04
CPFF(local)	6.784	294.95

REFERENCES

[1] R. Allen and S. Blackman, "Implementation of an angle-only tracking filter," in *Proc. of SPIE, Signal and Data Processing of Small Targets*, vol. 1481, Orlando, FL., USA, Apr. 1991.

[2] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, 1999.

[3] M. Arulampalam, S. Maskell, N. Gordon and T. Clapp, "A tutorial on particle filters for online nonlinear/ non-Gaussian Bayesian tracking," *IEEE Trans. on Signal Processing*, vol. 50, No. 2, pp. 174–188, Feb. 2002.

[4] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, John Wiley & Sons, 2001.

[5] K. Bell, and L. Stone. "Implementation of the homotopy particle filter in the JPDA and MAP-PF multi-target tracking algorithms," in *Proc. FUSION*, Salamnca, Spain, Jul. 2014.

[6] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, 1999.

[7] S. Blackman, T. White, B. Blyth, and C. Durand, "Integration of passive ranging with multiple hypothesis tracking (MHT) for application with angle-only measurements," in *Proc. of SPIE*, vol. 7698, Orlando, FL., USA, 2010.

[8] T. Bréhard and J-P. Le Cadre, "Closed-form posterior Cramér-Rao bound for a manoeuvring target in the bearings only tracking context using best-fitting Gaussian distribution," in *Proc. FUSION*, Florence, Italy, Jul. 2006.

[9] S. Choi, P. Willet, F. Daum and J. Huang, "Discussion and application of homotopy filter," in *Proc. SPIE Signal Processing, Sensor Fusion, and Target Recognition*, vol. 8050, Orlando, FL., USA, May 2011.

[10] F. Daum and J. Huang, "Angle only tracking with particle flow filters," in *Proc. of SPIE Signal and Data Processing of Small Targets*, vol. 8137, San Diego, CA, USA Aug. 2011.

[11] F. Daum and J. Huang, "How to avoid normalization of particle flow for nonlinear filters, Bayesian decisions, and transport", in *Proc. SPIE Defense and Security*, International Society for Optics and Photonics, Baltimore, MD, USA May 2014.

[12] F. Daum and J. Huang, "Nonlinear filters with log-homotopy," in *Proc. SPIE Signal and Data Processing of Small Targets*, vol. 6699, San Diego, CA., USA, 2007.

[13] F. Daum, J. Huang, and A. Noushin, "Exact particle flow for nonlinear filters," in *Proc. SPIE Signal Processing, Sensor Fusion, and Target Recognition*, vol. 7697, Orlando, FL., USA, Apr. 2010.

[14] F. Daum and J. Huang, "Particle flow for nonlinear filters," in *Proc. ICASSP*, Prague, Czech Republic, May 2011.

[15] F. Daum, "Book review: integrated tracking, classification and sensor management", (Edited by M. Mallick, V. Krishnamurthy, and B. Vo, Wiley-IEEE, Dec. 2012), *IEEE Aerospace and Electronic Systems Magazine*, vol. 29, no. 9, Sep. 2014, pp. 42–44.

[16] T. Ding and M. J. Coates, "Implementation of the Daum-Huang exact-flow particle filter," *Proc. IEEE SSP*, Ann-Arbor, MI., USA, Aug. 2012.

[17] G. Evensen, "The ensemble Kalman filter: theoretical formulation and practical implementation", *Ocean Dynamics*, vol. 53, no. 4, Nov. 2003, pp. 343–367.

[18] G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics", *J. Geophys. Res*, vol. 99. no. C5, May 1994, pp. 10143–10162.

[19] A. Gelb, Editor, *Applied Optimal Estimation*, MIT Press, 1974.

[20] S. Gillijns, O. Mendoza, J. Chandrasekar, B. De Moor, D. Bernstein, and A. Ridley, "What is the ensemble Kalman filter and how well does it work?", in *Proc. American Control Conf.*, Minneapolis, MN., USA, Jun. 2006.

[21] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/ non-Gaussian Bayesian state estimation," *Radar and Signal Processing, IEE Proceedings F*, vol. 140, no. 2, Apr. 1993, pp. 107–113.

[22] P. Gurfil and N. Kasdin, "Two-step optimal estimator for three dimensional target tracking," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 41, no. 3, Jul. 2005, pp. 780–793.

[23] S. Hammel and V. Aidala, "Utilization of modified polar coordinates for bearings-only tracking", *IEEE Trans. on Automatic Control*, vol. 28, no. 3, Mar. 1983, pp. 283–294.

[24] M. Hernandez, "Performance bounds for target tracking: computationally efficient formulations and associated applications", Ch. 7, in *Integrated Tracking, Classification, and Sensor Management: Theory and Applications*, Eds. M. Mallick, V. Krishnamurthy, and B. Vo, Wiley/ IEEE, Dec. 2012, pp. 255–309.

[25] R. Horn and C. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, 1991.

[26] S. Julier, J. Uhlmann and H. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Trans. on Automatic Control*, vol. 45, no. 3, Mar. 2000, pp. 477–482.

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

[28] R. Karlsson and F. Gustafsson, "Range estimation using angle-only target tracking with particle filters," in *Proc. American Control Conf.*, Arlington, VA., USA, Jun. 2001.

[29] M. Khan and M. Ulmke, "Non-linear and non-Gaussian state estimation using log-homotopy based particle flow filters, " in *Proc. Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Bonn, Germany, Oct. 2014.

[30] X. Li and V. Jilkov, " A survey of maneuvering target tracking, part VIc: approximate nonlinear density filtering in discrete time," in *Proc. SPIE, Signal and Data Processing of Small Targets* , vol. 8393, Baltimore, MD., USA, Apr. 2012.

[31] M. Mallick, L. Mihaylova, S. Arulampalam, and Y. Yan, "Angle-only Filtering in 3D Using Modified Spherical and Log Spherical Coordinates," in *Proc. FUSION*, Chicago, IL., USA, Jul. 2011.

[32] M. Mallick, M. Morelande , L. Mihaylova, S. Arulampalam, and Y. Yan, "Comparison of angle-only filtering algorithms in 3D using Cartesian and modified spherical coordinates," in *Proc. FUSION*, Singapore, Jul. 2012.

[33] M. Mallick, M. Morelande, L. Mihaylova, S. Arulampalam, and Y. Yan, Angle-only filtering in three dimensions, Ch. 1, in *Integrated Tracking, Classification, and Sensor Management: Theory and Applications*, Eds. M. Mallick, V. Krishnamurthy, and B. Vo, Ed., Wiley/IEEE, pp. 3–42, Dec. 2012, available at http://media.wiley.com/product_data/excerpt/59/04706390/0470639059-297.pdf.

[34] S. Ponda, *Trajectory optimization for target localization using small unmanned aerial vehicles*, M. S. Thesis, MIT, 2008.

[35] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter*, Artech House, 2004.

[36] P. Robinson, "Modified spherical coordinates for radar," *Proc. of the AIAA Guidance, Navigation and Control Conf.*, Scottsdale, AZ, pp. 55–64, Aug. 1-3, 1994.

[37] P. Sakov and P. Oke, 2008, "Implications of the form of the ensemble transformation in the ensemble square root filters", *Monthly Weather Review*, vol. 136, no. 3, Mar. 2008, pp. 1042–1053.

[38] D. Stallard, "An angle-only tracking filter in modified spherical coordinates," *Proc. of the AIAA Guidance, Navigation and Control Conf.*, Monterey, CA., USA, Aug. 1987, pp. 542–550.

[39] D. Stallard, "Angle-only tracking filter in modified spherical coordinates," *Journal of Guidance, Control, and Dynamics*, vol. 14, no. 3, May 1991, pp. 694–696.

[40] P. Tichavsky, C. Muravchik, and A. Nehorai, "Posterior Cramér- Rao bounds for discrete-time nonlinear filtering" *IEEE Trans. on Signal Processing*, vol. 46, no. 5, May 1998, pp. 1386–1396.

[41] M. Tippett, J. Anderson, C. Bishop, T. M. Hamill, and J. Whitaker, "Ensemble square root filters," *Monthly Weather Review*, vol. 131, no. 7, Jul. 2003, pp. 1485–1490.