

RESEARCH

Open Access



A novel distributed fusion algorithm for multi-sensor nonlinear tracking

Jingxian Liu¹, Zulin Wang^{1,2†} and Mai Xu^{1*†}

Abstract

The covariance intersection (CI), especially with feedback structure, can be easily combined with nonlinear filters to solve the distributed fusion problem of multi-sensor nonlinear tracking. However, this paper proves that the CI algorithm is sub-optimal, thus degrading the fusion accuracy. To avoid such an issue, a novel distributed fusion algorithm, namely Monte Carlo Bayesian (MCB) algorithm, is proposed. First, it builds a distributed fusion architecture based on the Bayesian tracking framework. Then, the Monte Carlo sampling is incorporated into this architecture to form a feasible solution to nonlinear tracking. Finally, the simulation results verify that our MCB algorithm advances the state-of-the-art distributed fusion of nonlinear tracking.

Keywords: Bayesian tracking framework, Monte Carlo sampling, Covariance intersection

1 Introduction

Distributed fusion [1] refers to combining the information of decentralized sensors [2, 3], in which the observation information of each sensor is processed independently. In comparison with centralized fusion, the distributed fusion can significantly save time and storage resources in the fusion center. As for distributed fusion, the conventional solution is Bar-Shalom-Campo algorithm [4]. Later, Chang et al. [5] proved that this algorithm is not optimal in terms of the root-mean-square-error (RMSE), and they further presented an optimal fusion algorithm. Nevertheless, this algorithm can only be used in the scenario of two sensors fusion. Hence, Li et al. [6] proposed a unified fusion architecture, which can be used for two and more sensors. However, all aforementioned algorithms are designed with linear state space models (SSM), which results in the inferior fusion performance of nonlinear tracking.

Recently, some intelligent techniques, such as the adaptive fuzzy backstepping control technique [7] and the adaptive neural network technique [8] with nonlinear model predictive control, are proposed to advance the application of nonlinear models in multi-sensors or sensor

network control [9]. Hence, the optimal distributed fusion with nonlinear model becomes one of the most important direction of signal processing. To our best knowledge, the most effective algorithms for fusion are based on covariance intersection (CI) [10, 11]. These CI-based algorithms can be easily combined with nonlinear filters to form the state-of-the-art solutions to nonlinear problems, such as UKF-SCI [12] and DPF-ICI [13]. The former is more accurate because the unscented Kalman filter (UKF) performs better in normally nonlinear tracking, while the latter has an advantage in non-Gaussian scenarios benefitting from the particle filter (PF). However, these CI-based algorithms require the computation on additional fractional powers to calculate the fusion results [14], which causes an increment of estimation error. Although many applications utilize a feedback structure¹ to improve the fusion accuracy [15], this paper proves that the sub-optimality caused by the increment of estimation error still exists in CI. To our best knowledge, few algorithms are proposed to overcome this sub-optimality. Hence, the present study is motivated to achieve a fusion algorithm which can outperform the CI algorithm. In summary, the novelty of this paper is that our algorithm overcomes the sub-optimality problem in covariance intersection (CI) fusion, which has not been addressed in the previous work.

*Correspondence: maixu@buaa.edu.cn

†Equal contributors

¹School of Electrical and Information Engineering, Beihang University, XueYuan Road No.37, Haidian District, 100191 Beijing, China

Full list of author information is available at the end of the article

In this paper, a Monte Carlo Bayesian (MCB) algorithm is proposed to overcome the disadvantage of the sub-optimality in CI algorithm, such that the fusion performance can be improved. Specifically, a distributed fusion architecture is designed based on the Bayesian tracking framework (BTF). This algorithm utilizes the law of total probability to form the distributed architecture, thus avoiding the increment of estimation error. Furthermore, the Monte Carlo sampling [16] is incorporated into our distributed architecture. This sampling method offers a direct approximate inference of the expectation of the target function with respect to a probability distribution [17]. Benefiting from this sampling method, the intractable problem of nonlinear estimation in BTF is circumvented by means of the approximation algorithm with sampling particles. Finally, based on the approximation, the fusion results are achieved in the form of the mean and variance of estimation at each step. The meanings of notation in the paper are listed in Table 1. The contributions of this work are listed as follows,

- A novel distributed fusion architecture is proposed, which makes full use of the information of different sensors to produce robust fusion estimation.

Table 1 Notation list

Notation	Meaning of the notation
\mathbf{x}_k	The state at time step k
\mathbf{P}_k	The variance of state at time step k
i	The i -th sensor
I	The number of sensors
n	The n -th particle
N	The number of particles
\mathbf{X}_k^n	The n -th particle of state at time step k
\mathbf{z}_k^i	The observation of the i -th sensor at time step k
$\mathbf{z}_{1:k}^f$	The observation set of all sensors from time step 1 to k
$p(\cdot)$	The transition distribution of state
$q(\cdot)$	The likelihood of state
$h^i(\cdot)$	The observation function of the i -th sensor
\mathbf{Q}	The variance of observation noise
$\delta(\cdot)$	The Dirac delta function
C	The normalized constant
F	The transition function
\mathbf{n}_{k-1}	The transition noise at time $k-1$
$d_{x,k}, d_{y,k}$	The distances along x and y direction at time step k
$v_{x,k}, v_{y,k}$	The velocity along x and y direction at time step k
s_T	The sampling interval
n_d, n_v	The transition noise of distance and velocity with variance σ_d^2 and σ_v^2
n_θ, n_r	The observation noise of azimuth and distance with variance σ_θ^2 and σ_r^2

- A MCB algorithm is developed by means of the Monte Carlo sampling, which solves the nonlinear fusion problem based on the numerical approximation.

2 Distributed fusion architecture based on BTF

In this section, our distributed fusion architecture is developed based on BTF. Let us consider the I sensors fusion scenario with feedback structure. The state and the i th observation sequences are denoted as $\{\mathbf{x}_k; k \in \mathbb{N}, \mathbf{x}_k \in \mathbb{R}^{d_x}\}$ and $\{\mathbf{z}_k^i; k \in \mathbb{N}, \mathbf{z}_k^i \in \mathbb{R}^{d_{z^i}}\}$, where k is the time step, $i = \{1, 2, \dots, I\}$ is the number of sensors, d_x and d_{z^i} are the dimensions of the state and the i -th observation vectors, respectively. Their relationship can be represented by the SSM as follows,

$$\text{Transition equation : } p(\mathbf{x}_k | \mathbf{x}_{k-1}), \quad k \geq 0, \quad (1)$$

$$\text{Observation equation : } q(\mathbf{z}_k^i | \mathbf{x}_k), \quad k \geq 0, \quad (2)$$

where $p(\cdot)$ is the transition distribution, and $q(\cdot)$ is the likelihood of \mathbf{x}_k in the i -th sensor. According to the BTF, the fusion process is to obtain the posterior distribution $p(\mathbf{x}_k | \mathbf{z}_{1:k}^f)$ based on the SSM, where $\mathbf{z}_{1:k}^f = \{\mathbf{z}_{1:k}^i\}_{i=1}^I$ is the integrated observation set combining all the sensors information from step 1 to k . Notice that, in the feedback structure, we normally have $\mathbf{z}_{1:k}^i = \{\mathbf{z}_k^i, \mathbf{z}_{1:k-1}^i\}$ because the prior observation information has been shared between sensors. Then, the fusion is calculated iteratively with two steps: the prediction and update steps.

The prediction step:

$$p(\mathbf{x}_k | \mathbf{z}_{1:k-1}^f) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}^f) d\mathbf{x}_{k-1}. \quad (3)$$

The update step:

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}^f) = \frac{q(\mathbf{z}_k^i | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{z}_{1:k-1}^f)}{p(\mathbf{z}_k^i | \mathbf{z}_{1:k-1}^f)}, \quad (4)$$

where

$$p(\mathbf{z}_k^i | \mathbf{z}_{1:k-1}^f) = \int q(\mathbf{z}_k^i | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{z}_{1:k-1}^f) d\mathbf{x}_k \quad (5)$$

is the marginal likelihood.

In normal cases, the current observation from each sensor is independent of others. Thus, the posterior distribution of (4) can be recalculated according to the law of total probability as follows,

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}^f) \propto p(\mathbf{x}_k | \mathbf{z}_{1:k-1}^f) \prod_{i=1}^I q(\mathbf{z}_k^i | \mathbf{x}_k). \quad (6)$$

Then, our distributed architecture is obtained by utilizing (6) to decompose the computation of (4) to each sensor, which is summarized in Fig. 1.

Figure 1 shows that, in the fusion process of time step k , the fusion center makes a prediction with (3). Then, the information of prediction is sent to each distributed sensor, in which the likelihood is calculated. Afterwards, each sensor sends the likelihood back to the fusion center to yield the final result with (6). Finally, the fusion result feeds back to the fusion center as the prior information of the next step.

3 Sub-optimality in CI algorithm with feedback structure

This section analyzes the sub-optimality of the existing CI algorithm with feedback structure, by comparing it with the aforementioned distributed fusion in Section 2. This paper only concentrates on the most common scenarios that the distribution in SSM is Gaussian [18]. Without loss of generality, the fusion with I sensors is considered, where $I \geq 2$. According to [19], for any $w \in (0, 1)$, the posterior distribution of two sensors fusion in CI algorithm holds:

$$p(\mathbf{x}_k | \{\mathbf{z}_{1:k}^1, \mathbf{z}_{1:k}^2\}) \propto p(\mathbf{x}_k | \mathbf{z}_{1:k}^1)^\omega p(\mathbf{x}_k | \mathbf{z}_{1:k}^2)^{1-\omega}. \quad (7)$$

Lemma 1. Let $\mathbf{x}_k \in \mathbb{R}^{d_x}$, $\mathbf{z}_k^i \in \mathbb{R}^{d_{z_i}}$, where $k \in \mathbb{N}$; $i = \{1, 2, \dots, I\}$, $I \in \mathbb{N}$, $I \geq 2$; d_x and d_{z_i} are the dimensions of vectors \mathbf{x}_k and \mathbf{z}_k^i . Let $\mathbf{z}_{1:k}^i = \{\mathbf{z}_{1:k}^i\}_{i=1}^I$, where $\mathbf{z}_{1:k}^i = \{\mathbf{z}_k^i, \mathbf{z}_{1:k-1}^i\}$. Given (7), the posterior distribution $p(\mathbf{x}_k | \mathbf{z}_{1:k}^f)$ can be derived as follows,

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}^f) \propto p(\mathbf{x}_k | \mathbf{z}_{1:k-1}^f) \prod_{i=1}^I q(\mathbf{z}_k^i | \mathbf{x}_k)^{\omega_i}, \quad (8)$$

where $w_i \in (0, 1)$ and $\sum_{i=1}^I w_i = 1$.

Proof. We decompose $p(\mathbf{x}_k | \mathbf{z}_{1:k}^f)$ into a product form according to (7) as follows

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}^f) = p(\mathbf{x}_k | \{\mathbf{z}_{1:k}^i\}_{i=1}^I) \propto \left(\prod_{i=1}^I p(\mathbf{x}_k | \mathbf{z}_{1:k}^i)^{\omega'_i \prod_{j=0}^{i-1} (1-\omega'_j)} \right) p(\mathbf{x}_k | \mathbf{z}_{1:k}^I)^{\prod_{i=1}^{I-1} (1-\omega'_i)}, \quad (9)$$

where $\omega'_0 \equiv 0$, and $w'_i \in (0, 1)$, $i = 1, 2, \dots, I$. Then, (9) can be rewritten as

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}^f) \propto \prod_{i=1}^I p(\mathbf{x}_k | \mathbf{z}_{1:k}^i)^{\omega_i}, \quad (10)$$

where $\omega_i \triangleq \omega'_i \prod_{j=0}^{i-1} (1-\omega'_j)$, $i = 1, 2, \dots, I-1$, and $\omega_I \triangleq \prod_{i=1}^I (1-\omega'_i)$. It can be easily known that $w_i \in (0, 1)$ and $\sum_{i=1}^I w_i = 1$. Based on (10), we finally have

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}^f) \propto \prod_{i=1}^I p(\mathbf{x}_k | \mathbf{z}_k^i, \mathbf{z}_{1:k-1}^f)^{\omega_i} \propto p(\mathbf{x}_k | \mathbf{z}_{1:k-1}^f) \prod_{i=1}^I q(\mathbf{z}_k^i | \mathbf{x}_k)^{\omega_i}. \quad (11)$$

This completes the proof of Lemma 1.

In the feedback structure, Lemma 1 provides a generalized form of posterior distribution $p(\mathbf{x}_k | \mathbf{z}_{1:k}^f)$ calculated in CI fusion. As seen from (8) and (6), the CI algorithm adds a fractional power ω_i to each likelihood. \square

Lemma 2. Assuming that likelihoods obey Gaussian distribution, i.e., $q(\mathbf{z}_k^i | \mathbf{x}_k) = \mathcal{N}(\mathbf{z}_k^i; h^i(\mathbf{x}_k), \sigma_i^2)$ for each $i = \{1, 2, \dots, I\}$, where $h^i(\cdot)$ is the mapping from \mathbb{R}^{d_x} to $\mathbb{R}^{d_{z_i}}$, and σ_i^2 is the corresponding variance. Then, the variance of likelihood in each sensor becomes σ_i^2 / ω_i in CI algorithm.

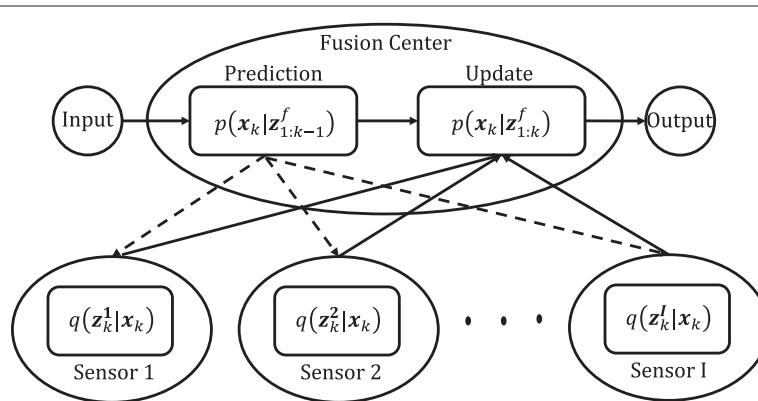


Fig. 1 The distributed fusion architecture. The figure describes the fusion architecture in which sensors and fusion center are linked with information flow

Proof. According to Lemma 1, the posterior distribution of CI is calculated with (8) as

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}^f) \propto p(\mathbf{x}_k | \mathbf{z}_{1:k-1}^f) \prod_{i=1}^I \exp\left(-\frac{(\mathbf{z}_k^i - h^i(\mathbf{x}_k))^2}{2\sigma_i^2/\omega_i}\right). \quad (12)$$

Hence, the variance of each likelihood in CI fusion has turned into σ_i^2/ω_i in the process of calculating the posterior distribution. This completes the proof of Lemma 2. \square

Since there exists

$$\omega_i \in (0, 1) \Rightarrow \sigma_i^2/\omega_i > \sigma_i^2,$$

ω_i increases the variance of likelihood in CI fusion. Hence, compared with (6) in the distributed fusion of Section 2, the estimation (8) calculated in CI contains more uncertainty under the same observations. Such uncertainty leads to the increment of estimation error. As a result, the sub-optimality still exists in CI fusion with feedback structure.

4 Monte Carlo Bayesian algorithm

To avoid the increment of estimation error, this paper utilizes the distributed fusion architecture in Section 2 to develop the fusion algorithm. In the most common applications, the observation is nonlinear [20]. Therefore, the posterior distributed in (6) cannot be simply calculated by joint Gaussian method with the procedure of squares completion [21]. To solve this problem, our MCB algorithm is proposed. This algorithm incorporates the Monte Carlo sampling into the distributed fusion architecture, by which the posterior distribution can be directly approximated with particles.

4.1 Monte Carlo sampling in MCB algorithm

At time k , N independent random particles are drawn from prediction distribution $p(\mathbf{x}_k | \mathbf{z}_{1:k-1}^f)$ in (3):

$$\{\mathbf{X}_k^n \sim p(\mathbf{x}_k | \mathbf{z}_{1:k-1}^f)\}_{n=1}^N, \quad (13)$$

where \mathbf{X}_k^n is the n -th particle drawn from $p(\mathbf{x}_k | \mathbf{z}_{1:k-1}^f)$. According to the approximation inference method of numerical sampling [17], the prediction distribution $p(\mathbf{x}_k | \mathbf{z}_{1:k-1}^f)$ can be approximated as follows,

$$p(\mathbf{x}_k | \mathbf{z}_{1:k-1}^f) \approx \frac{1}{N} \sum_{n=1}^N \delta_{\mathbf{X}_k^n}(\mathbf{x}_k). \quad (14)$$

In (14), $\delta_{\mathbf{X}_k^n}(\cdot)$ denotes the Dirac delta function,

$$\delta_{\mathbf{X}_k^n}(x) = \begin{cases} 1, & x = \mathbf{X}_k^n, \\ 0, & x \neq \mathbf{X}_k^n. \end{cases} \quad (15)$$

According to (14), the posterior distribution in (6) can be approximated as,

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}^f) \approx \left[\frac{C}{N} \sum_{n=1}^N \delta_{\mathbf{X}_k^n}(\mathbf{x}_k) \right] \prod_{i=1}^I q(\mathbf{z}_k^i | \mathbf{x}_k) = \frac{C}{N} \sum_{n=1}^N \left[\prod_{i=1}^I q(\mathbf{z}_k^i | \mathbf{x}_k) \right] \delta_{\mathbf{X}_k^n}(\mathbf{x}_k), \quad (16)$$

where C is the normalization constant.

4.2 Fusion with MCB algorithm

According to the distributed fusion architecture in section 2, the estimation of state \mathbf{x}_k is calculated iteratively by (3) and (6).

First, in the fusion center, at time step k , given the previous state estimation with the mean $\hat{\mathbf{x}}_{k-1}$ and variance \mathbf{P}_{k-1} , the unscented sigma point set χ_{k-1} is calculated as follows,

$$\chi_{k-1} = [\hat{\mathbf{x}}_{k-1}, \hat{\mathbf{x}}_{k-1} \pm \sqrt{(I_x + \lambda)\mathbf{P}_{k-1}}]. \quad (17)$$

In (17), $\chi_{i,k-1} \in \chi_{k-1}$ denotes the i th point in the sigma point set, and λ is the scaling parameter [22]. Then, the prediction (3) is calculated in the form of mean $\bar{\mathbf{x}}_{k|k-1}$ and variance $\mathbf{P}_{k|k-1}$ [22] as

$$\bar{\mathbf{x}}_{k|k-1} = \sum_{i=0}^{2I_x} W_i^m f(\chi_{i,k-1}), \quad (18)$$

$$\mathbf{P}_{k|k-1} = \mathbf{Q} + \sum_{i=0}^{2I_x} W_i^c [f(\chi_{i,k-1}) - \bar{\mathbf{x}}_{k|k-1}][f(\chi_{i,k-1}) - \bar{\mathbf{x}}_{k|k-1}]^T, \quad (19)$$

where $f(\cdot)$ is the state transition function, \mathbf{Q} is the variance of corresponding noise, $W_i^m \in \mathbf{W}^m$ and $W_i^c \in \mathbf{W}^c$ are unscented transformation parameters calculated before estimation [22]. Then, N particles are drawn from the Gaussian approximation of prediction:

$$\{\mathbf{X}_k^n \sim \mathcal{N}(\mathbf{x}_k; \bar{\mathbf{x}}_{k|k-1}, \mathbf{P}_{k|k-1})\}_{n=1}^N.$$

Second, the particles are sent to each sensor node. After achieving the observation \mathbf{z}_k^i in the i th sensor, the likelihood is approximated by the set: $\{q(\mathbf{z}_k^i | \mathbf{X}_k^n)\}_{n=1}^N$. Then, each sensor sends its own approximated likelihood back to the fusion center.

Finally, the fusion center receives all the approximated likelihoods, and calculates the fusion results in the form of the mean and variance with (16) as follows,

$$\begin{aligned}
\bar{\mathbf{x}}_k &= \int \mathbf{x}_k p(\mathbf{x}_k | \mathbf{z}_{1:k-1}^f) d\mathbf{x}_k \\
&\approx \int \mathbf{x}_k \left\{ \frac{C}{N} \sum_{n=1}^N \left[\prod_{i=1}^I q(\mathbf{z}_k^i | \mathbf{x}_k) \right] \delta_{X_k^n}(\mathbf{x}_k) \right\} d\mathbf{x}_k \\
&= \frac{C}{N} \sum_{n=1}^N \left[\prod_{i=1}^I q(\mathbf{z}_k^i | X_k^n) \right] X_k^n, \quad (20)
\end{aligned}$$

$$\begin{aligned}
\mathbf{P}_k &= \int (\mathbf{x}_k - \bar{\mathbf{x}}_k) (\mathbf{x}_k - \bar{\mathbf{x}}_k)^T p(\mathbf{x}_k | \mathbf{z}_{1:k-1}^f) d\mathbf{x}_k \\
&\approx \int (\mathbf{x}_k - \bar{\mathbf{x}}_k) (\mathbf{x}_k - \bar{\mathbf{x}}_k)^T \left\{ \frac{C}{N} \sum_{n=1}^N \left[\prod_{i=1}^I q(\mathbf{z}_k^i | \mathbf{x}_k) \right] \delta_{X_k^n}(\mathbf{x}_k) \right\} d\mathbf{x}_k \\
&= \frac{C}{N} \sum_{n=1}^N \int \left\{ \prod_{i=1}^I [q(\mathbf{z}_k^i | \mathbf{x}_k)] (\mathbf{x}_k - \bar{\mathbf{x}}_k) (\mathbf{x}_k - \bar{\mathbf{x}}_k)^T \delta_{X_k^n}(\mathbf{x}_k) \right\} d\mathbf{x}_k \\
&= \frac{C}{N} \sum_{n=1}^N \left[\prod_{i=1}^I q(\mathbf{z}_k^i | X_k^n) \right] [X_k^n - \bar{\mathbf{x}}_k] [X_k^n - \bar{\mathbf{x}}_k]^T. \quad (21)
\end{aligned}$$

5 Simulation results

In this section, the simulation results are provided to validate the performance of our MCB algorithm. In our simulation, a classic two-dimensional (2-D) fusion scenario with the nonlinear observation of one active and two passive radars is considered. In such a scenario, the fusion process is simulated by tracking a single target under several cases. For each case, different noise and kinematic models of transition equation are applied. Finally, 100 Monte Carlo simulations have been run for each case, and the fusion performance of nonlinear tracking is compared between our MCB algorithm and the UKF-SCI algorithm [12] with feedback structure. Note that, in this paper, we concentrate on Gaussian tracking scenarios, in which the UKF-SCI outperforms the DPF-ICI [13].

5.1 Simulation setup

In the common radar system, the transition Eq. (1) is considered as

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{n}_{k-1}, \quad (22)$$

where $\mathbf{x}_k = [d_{x,k}, d_{y,k}, v_{x,k}, v_{y,k}]^T$ is the column vector, indicating the 2-D distance and velocity of a single target in the $x - y$ plane at time step k . \mathbf{F} is the state transition matrix with sampling interval s_T . In this paper, three state transition matrices are taken into account for three kinematic models: constant-velocity (CV) and constant-turn (CT) with known turn rates $\omega = 2.5^\circ/s$ and $\omega = 5^\circ/s$ [23];

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & s_T & 0 \\ 0 & 1 & 0 & s_T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (23)$$

for CV model and

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & \frac{\sin(\omega s_T)}{\omega} & \frac{\cos(\omega s_T) - 1}{\omega} \\ 0 & 1 & \frac{1 - \cos(\omega s_T)}{\omega} & \frac{\sin(\omega s_T)}{\omega} \\ 0 & 0 & \cos(\omega s_T) & -\sin(\omega s_T) \\ 0 & 0 & \sin(\omega s_T) & \cos(\omega s_T) \end{bmatrix} \quad (24)$$

for CT model [24]. The trajectories of three kinematic models in our scenario are shown in Fig. 2.

In addition, $\mathbf{n}_k = [n_d, n_d, n_v, n_v]^T$ represents the state transition noise in distance and velocity. For fair comparison, this paper utilizes the same noise for different models which obeys the Gaussian distribution, i.e., $n_d \sim \mathcal{N}(n_d; 0, \sigma_d^2)$ and $n_v \sim \mathcal{N}(n_v; 0, \sigma_v^2)$. Here, σ_d and $\sigma_v = 2\sigma_d/s_T$ are standard deviations of transition noise of distance and velocity, respectively.

In our simulation, the observation equation in (2) is replaced by active and passive radar observation equations defined as,

$$\text{Active : } \mathbf{z}_k = \begin{bmatrix} \arctan \frac{d_{y,k}}{d_{x,k}} \\ \sqrt{d_{x,k}^2 + d_{y,k}^2} \end{bmatrix} + \begin{bmatrix} n_\theta \\ n_r \end{bmatrix}, \quad (25)$$

$$\text{Passive : } \mathbf{z}_k = \arctan \frac{d_{y,k}}{d_{x,k}} + n_\theta, \quad (26)$$

where $n_\theta \sim \mathcal{N}(n_\theta; 0, \sigma_\theta^2)$, $n_r \sim \mathcal{N}(n_r; 0, \sigma_r^2)$; σ_θ and σ_r are standard deviations for tracking azimuth and distance. Then, we evaluate the performance of fusion with three radars, i.e., one active and two passive radars. Following [25], the aforementioned parameters are set as $s_T = 0.1(s)$, $\sigma_\theta = 0.0001(rad)$, $\sigma_r = 0.5(m)$, and $\sigma_d = \{0.25, 0.3, 0.35, 0.4\}(m)$.

The simulation setting may be applied to a composite guidance scenario [26], in which there are one active and two passive radars [27, 28]. Although we simplify the observation equation in a 2-D scenario, the simulation scenario is still suitable for the practical application of tracking and surveillance in network centric warfare (NCW) [29].

5.2 Evaluation

In our simulation, the single target tracking is performed with the aforementioned kinematic models. For each model, both MCB and UKF-SCI algorithms are utilized to fuse the information of three radars (one active and two passive) together, and obtain the final tracking result. Furthermore, in the tracking process of each model, the fusion performance is evaluated with four standard deviations of transition noise (i.e., $\{0.25, 0.3, 0.35, 0.4\}(m)$). Note that 100 Monte Carlo runs are applied in our simulation for each scenario, and the particle number N in our MCB algorithm is set to be 200.

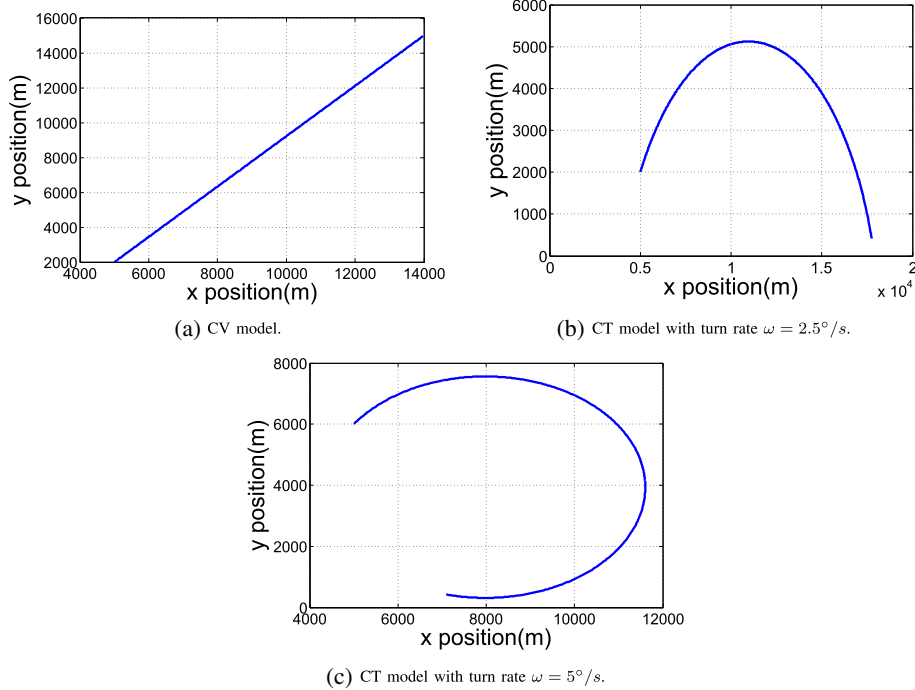


Fig. 2 The trajectories of target for three dynamic models. There are three sub-figures **a**, **b**, and **c** in this figure. **a** is the trajectory drew according CV model. **b** is the trajectory drew according CT model with turn rate $\omega = 2.5^\circ/s$. **c** is the trajectory drew according CT model with turn rate $\omega = 5^\circ/s$

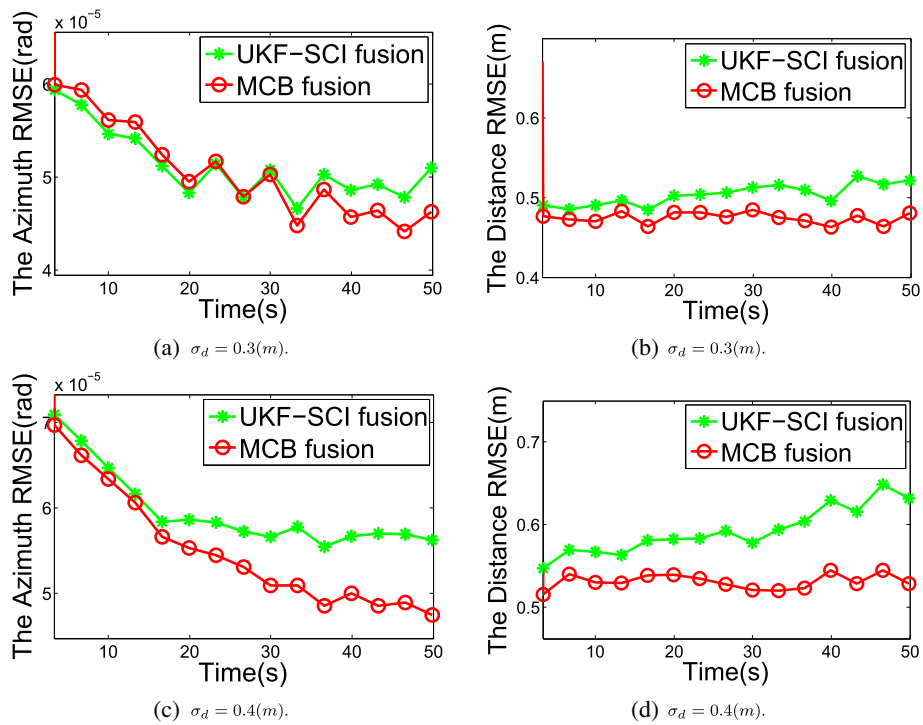


Fig. 3 The fusion RMSE of azimuth and distance for both MCB and UKF-SCI algorithms, with CV model. In the upper figures, $\sigma_d = 0.3(m)$, and in the bottom figures, $\sigma_d = 0.4(m)$. There are four sub-figures **a**, **b**, **c**, and **d** in this figure, in which, under the CV model, the changes of fusion RMSE of azimuth and distance are described along with time steps. Meanwhile, **a** and **b** show the comparison between MCB and UKF-SCI algorithms under the case of standard deviations of distance $\sigma_d = 0.3(m)$; **b** and **d** show the comparison between MCB and UKF-SCI algorithms under the case of standard deviations of distance $\sigma_d = 0.4(m)$

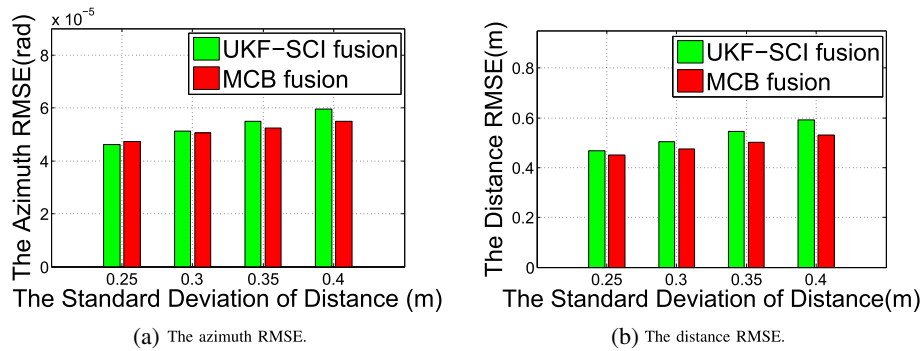


Fig. 4 The comparison of azimuth and distance RMSE at different standard deviations of distance, with CV model. There are two sub-figures **a** and **b** in this figure, in which, under the CV model, the changes of fusion RMSE of azimuth and distance are described along with different standard deviations of distance. The fusion RMSE in **a** and **b** is the mean of all results covering all fusion time steps

The simulation results of CV model are shown in Figs. 3 and 4. In Fig. 3, two scenarios with $\sigma_d = \{0.3, 0.4\}$ are selected to show the fusion root-mean-squared-error (RMSE) averaged over all 100 Monte Carlo simulations, along with the tracking time. Obviously, we can see from Fig. 3 that our MCB algorithm reduces the RMSE of both azimuth and distance, along with the tracking time.

Such RMSE reduction becomes larger, when the transition noise increases. Furthermore, in Fig. 4, we show the average RMSE (from tracking time 3 to 50 s) of MCB and UKF-SCI algorithms with all scenarios of four transition noises. As seen from this figure, our MCB algorithm outperforms the UKF-SCI algorithm in all cases except the azimuth case of $\sigma_d = 0.25$. In addition, the improvement

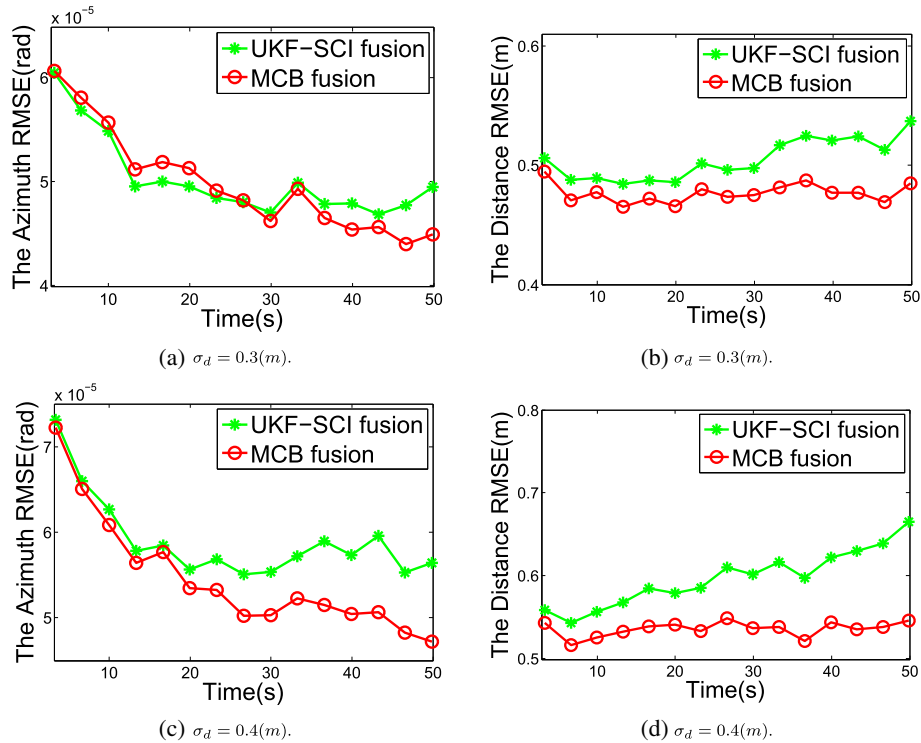


Fig. 5 The fusion RMSE of azimuth and distance for both MCB and UKF-SCI algorithms, with CT model of turn rate $2.5^\circ/s$. In the upper figures, $\sigma_d = 0.3(m)$, and in the bottom figures, $\sigma_d = 0.4(m)$. There are four sub-figures **a**, **b**, **c**, and **d** in this figure, in which, under the CT model of turn rate $\omega = 2.5^\circ/s$, the changes of fusion RMSE of azimuth and distance are described along with time steps. Meanwhile, **a** and **b** show the comparison between MCB and UKF-SCI algorithms under the case of standard deviations of distance $\sigma_d = 0.3(m)$; **b** and **d** show the comparison between MCB and UKF-SCI algorithms under the case of standard deviations of distance $\sigma_d = 0.4(m)$

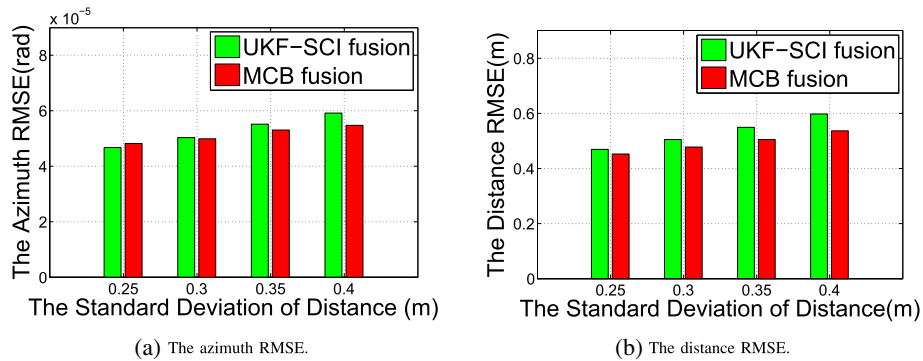


Fig. 6 The comparison of azimuth and distance RMSE at different standard deviations of distance, with CT model of turn rate $2.5^\circ/s$. There are two sub-figures **(a)** and **(b)** in this figure, in which, under the CT model of turn rate $2.5^\circ/s$, the changes of fusion RMSE of azimuth and distance are described along with different standard deviations of distance. The fusion RMSE in **(a)** and **(b)** is the mean of all results covering all fusion time steps

of our MCB algorithm increases, when the transition noise becomes larger.

The simulation results of CT model are shown in Figs. 5, 6, 7, and 8. Figures 5 and 6 show the fusion RMSE comparison of MCB and UKF-SCI algorithms with the known turn rate $\omega = 2.5^\circ/s$. Figures 7 and 8 show the same things with the known turn rate $\omega = 5^\circ/s$. All the results in these

figures are similar to the ones in Figs. 3 and 4. Hence, our MCB algorithm also outperforms the UKF-SCI algorithm with CT models. In addition, our MCB algorithm can perform better when the turn rate increases. The details are shown in Tables 2 and 3.

Tables 2 and 3 depict the proportions of reduced RMSE for different kinematic models and noises. As seen in

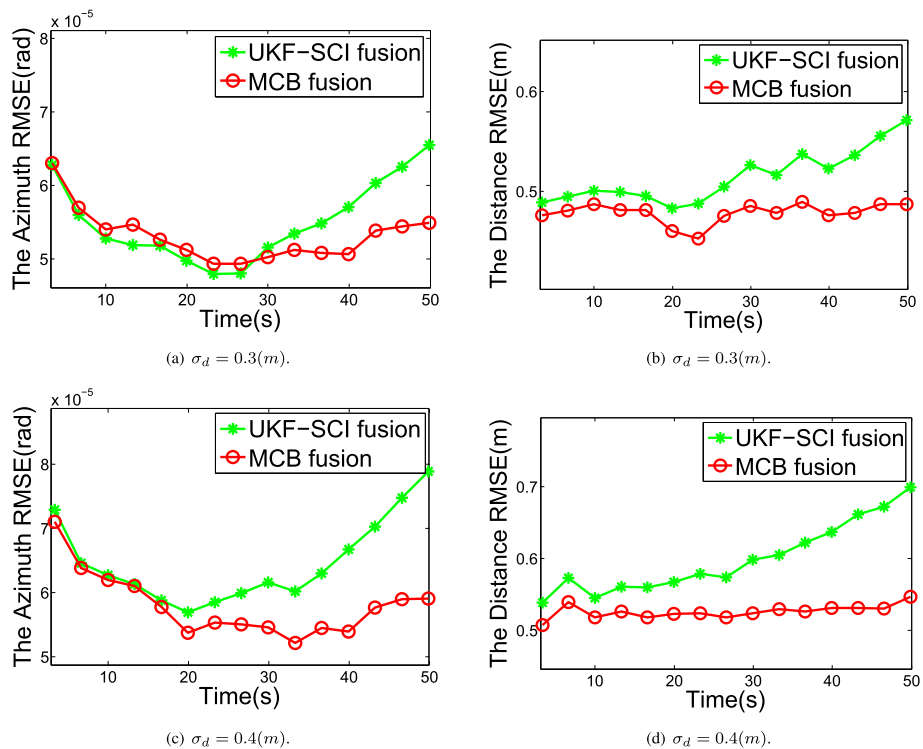
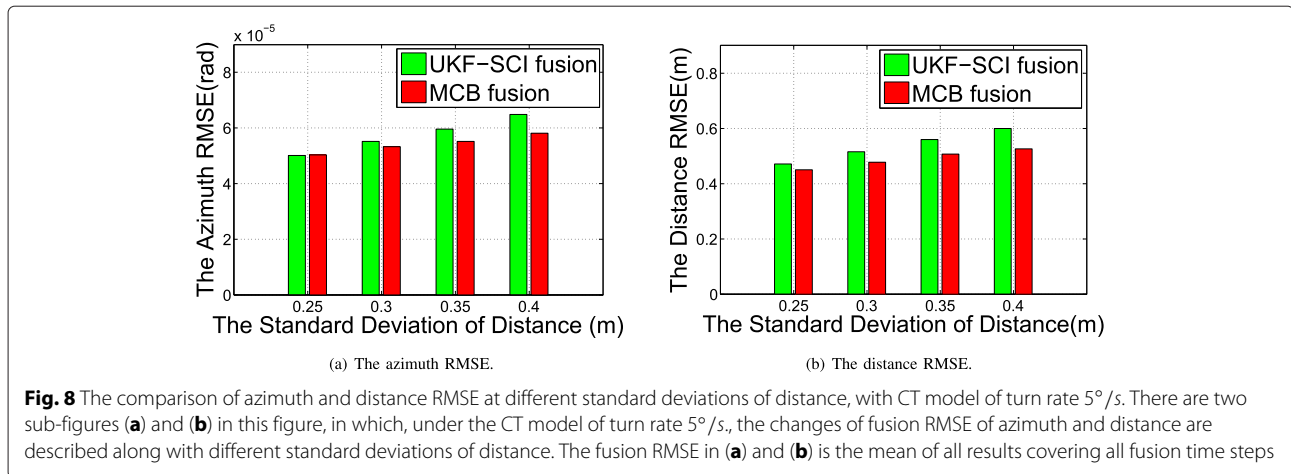


Fig. 7 The fusion RMSE of azimuth and distance for both MCB and UKF-SCI algorithms, with CT model of turn rate $5^\circ/s$. In the upper figures, $\sigma_d = 0.3(m)$, and in the bottom figures, $\sigma_d = 0.4(m)$. There are four sub-figures **a**, **b**, **c**, and **d** in this figure, in which, under the CT model of turn rate $\omega = 5^\circ/s$, the changes of fusion RMSE of azimuth and distance are described along with time steps. Meanwhile, **a** and **b** show the comparison between MCB and UKF-SCI algorithms under the case of standard deviations of distance $\sigma_d = 0.3(m)$; **c** and **d** show the comparison between MCB and UKF-SCI algorithms under the case of standard deviations of distance $\sigma_d = 0.4(m)$



these two tables, the reduction of RMSE becomes larger when the transition noise increases. This is consistent with the results shown in aforementioned figures. Moreover, the reduction of RMSE with CT model of turn rate 5°/s is largest among all three kinematic models. That means, our MCB algorithm has more advantage for high maneuvering targets.

In summary, based on the BTE, our MCB algorithm makes full use of the information of all observations, and fuses it to obtain more accurate estimation on target tracking. Hence, compared with the CI algorithm, there are two advantages in our MCB algorithm: (1) When the transition noise is large, the fusion RMSE of azimuth and distance is still small; (2) When the turn rate is large, the fusion RMSE of azimuth and distance is small as well. In other words, in the high maneuvering cases such as large transition noise and turn rate, our MCB outperforms the state-of-the-art CI algorithm, in terms of the fusion RMSE.

5.3 Computational complexity

In this section, the computational complexity of our MCB algorithm is analysed. According to Section 4.2, in the fusion center, the algorithm contains two steps for each iteration: the prediction and update steps. In the prediction step, according to (18) and (19), the mean and variance are calculated with a summed form, in which the computational complexity is proportional to

the dimension of tracking state. Moreover, N particles are drawn by sampling precess whose complexity is $O(N)$. In the update step, firstly, we need to multiple all I sensors information of each particle to form a fusion likelihood. The computational complexity of this process is $O(I)$. Then, the likelihood is used to compute the fusion results with the complexity being $O(N)$, according to (20) and (21). In summary, the computational complexity in the update step is $O(I \cdot N)$.

To further evaluate the computational complexity of our MCB algorithm, we have recorded the computational time of the prediction and update steps, respectively, for one iteration in the simulation. Specially, the computer used for the test is with Intel Core i7-3770 CPU at 3.4 GHz and 4 GB RAM. In the aforementioned tracking case in Sections 5.1 and 5.2, the dimension of tracking state is 4, the particle number is 200 and the sensor number is 3. Through the simulation, we found out that the prediction and update steps take around 0.563 and 2.3 ms for one iteration. In other word, we only need 2.863 ms to compute the tracking results in the fusion center at each iteration. Hence, this algorithm is fast enough to utilize in the fields of radar tracking and fusion.

6 Conclusions

In this paper, we have proposed a novel MCB algorithm to achieve the distributed fusion estimation of nonlinear tracking. First, the distributed fusion architecture is set

Table 2 The proportion of azimuth RMSE reduced by MCB algorithm over UKF-SCI, with four standard deviations of transition noise

	0.25(m)	0.3(m)	0.35(m)	0.4(m)
CV	-2.51 %	1.29 %	4.90 %	7.78 %
CT with turn rate of 2.5°/s	-3.19 %	0.88 %	4.10 %	7.43 %
CT with turn rate of 5°/s	-0.53 %	3.47 %	7.47 %	10.36 %

Table 3 The proportion of distance RMSE reduced by MCB algorithm over UKF-SCI, with four standard deviations of transition noise

	0.25(m)	0.3(m)	0.35(m)	0.4(m)
CV	3.55 %	5.80 %	8.02 %	10.37 %
CT with turn rate of 2.5°/s	3.51 %	5.57 %	8.06 %	10.25 %
CT with turn rate of 5°/s	4.54 %	7.06 %	9.42 %	12.24 %

up based on BTF. Second, the sub-optimality in CI algorithms is proved. Then, to solve the estimation problem of nonlinear tracking, the Monte Carlo sampling method is incorporated into the distributed architecture. Benefiting from this sampling method, the approximation of fusion results is obtained through random particles. Simulation results verify that our MCB algorithm outperforms the state-of-the-art CI algorithm.

In summary, there are three directions of the future work in our paper. (1) Our MCB algorithm only offers a distributed calculation on the update step. Hence, a total distributed fusion structure is needed to further reduce the computation and communication overhead. (2) The time-discrete SSM used in our paper is actually a special case of continuous-time SSM. Therefore, we can extend our method to the exponential tracking scenario, in which the filtering can be processed with partially unknown and uncertain transition probabilities [30–33] with Markovian jump system. (3) Unknown inputs which represent the faults can be added into SSM and the residuals are calculated [34]. Hence, the fault detection algorithms [35] and the fuzzy model [36] can also be incorporated into our fusion system to strengthen the reliability of fusion process in the future work.

Endnote

¹The feedback structure means a structure which feeds the tracking result back to each sensor as the prior knowledge for the next step.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

This work is a product of inputs and work of all the authors to the research concept and experiment.

Acknowledgements

This work was supported by the National Nature Science Foundation of China projects under Grants 61471022 and 61573037.

Received: 30 December 2015 Accepted: 6 May 2016

Published online: 18 May 2016

References

- DL Hall, J Llinas, An introduction to multisensor data fusion. *Proc. IEEE*. **85**(1), 6–23 (1997). doi:10.1109/5.554205
- S Xu, RC de Lamare, HV Poor, Adaptive link selection algorithms for distributed estimation. *EURASIP J. Adv. Signal Process.* **2015**(1), 1–22 (2015)
- SB Hassen, A Samet, An efficient central doa tracking algorithm for multiple incoherently distributed sources. *EURASIP J. Adv. Signal Process.* **2015**(1), 1–19 (2015)
- Y Bar-Shalom, L Campo, The effect of the common process noise on the two-sensor fused-track covariance. *Aerosp. Electron. Syst. IEEE Trans.* **AES-22**(6), 803–805 (1986). doi:10.1109/TAES.1986.310815
- KC Chang, RK Saha, Y Bar-Shalom, M Alford, in *Multisensor Fusion and Integration for Intelligent Systems, 1996. IEEE/SICE/RSJ International Conference On*. Performance evaluation of multisensor track-to-track fusion (IEEE, Washington, DC, 1996), pp. 627–632. doi:10.1109/MFI.1996.572239
- XR Li, Y Zhu, J Wang, C Han, Optimal linear estimation fusion .i. unified fusion rules. *Inf. Theory IEEE Trans.* **49**(9), 2192–2208 (2003). doi:10.1109/TIT.2003.815774
- T Wang, Y Zhang, J Qiu, H Gao, Adaptive fuzzy backstepping control for a class of nonlinear systems with sampled and delayed measurements. *IEEE Trans. Fuzzy Syst.* **23**(2), 302–312 (2015)
- T Wang, H Gao, J Qiu, A combined adaptive neural network and nonlinear model predictive control for multirate networked industrial process control. *IEEE Trans. Neural Netw. Learn. Syst.* **27**(99), 416–425 (2015)
- T Wang, H Gao, J Qiu, A combined fault-tolerant and predictive control for network-based industrial processes. *IEEE Trans. Ind. Electron.* **63**, 1–1 (2016)
- SJ Julier, JK Uhlmann, in *American Control Conference, 1997. Proceedings of the 1997*. A non-divergent estimation algorithm in the presence of unknown correlations, vol. 4 (IEEE, Albuquerque, NM, 1997), pp. 2369–23734. doi:10.1109/ACC.1997.609105
- M Reinhardt, B Noack, PO Arambel, UD Hanebeck, Minimum covariance bounds for the fusion under unknown correlations. *Signal Process. Lett. IEEE.* **22**(9), 1210–1214 (2015)
- Z Deng, P Zhang, W Qi, J Liu, Y Gao, Sequential covariance intersection fusion kalman filter. *Inf. Sci.* **189**, 293–309 (2012)
- O Hlinka, O Sluciak, F Hlawatsch, M Rupp, in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference On*. Distributed data fusion using iterative covariance intersection (IEEE, Florence, 2014), pp. 1861–1865
- SJ Julier, in *Target Tracking and Data Fusion: Algorithms and Applications, 2008 IET Seminar On*. Fusion without independence (IET, Birmingham, 2008), pp. 1–1
- S Mori, KC Chang, CY Chong, in *Information Fusion (FUSION), 2012 15th International Conference On*. Comparison of track fusion rules and track association metrics (IEEE, 2012), pp. 1996–2003
- A Doucet, X Wang, Monte carlo methods for signal processing: a review in the statistical signal processing context. *Signal Process. Mag. IEEE.* **22**(6), 152–170 (2005). doi:10.1109/MSP.2005.1550195
- CM Bishop, *Pattern recognition and machine learning (information science and statistics)*. (Springer, New York, 2006), pp. 523–524
- T Song, H Kim, D Musicki, Distributed (nonlinear) target tracking in clutter. *Aerosp. Electron. Syst. IEEE Trans.* **51**(1), 654–668 (2015)
- MB Hurley, in *Information Fusion, 2002. Proceedings of the Fifth International Conference On*. An information theoretic justification for covariance intersection and its generalization, vol. 1 (IEEE, Annapolis, MD, USA, 2002), pp. 505–5111. doi:10.1109/ICIF.2002.1021196
- Y He, JJ Xiu, JW Zhang, X Guan, *Radar Data Processing with Applications*. (Beijing: Publishing house of electronics industry, Beijing, 2006)
- A Doucet, S Godsill, C Andrieu, On sequential Monte Carlo sampling methods for Bayesian filtering. *Stat. Comput.* **10**(3), 197–208 (2000)
- R Van Der Merwe, A Doucet, N De Freitas, E Wan, in *NIPS*. The unscented particle filter (MIT press, Cambridge, MA, 2000), pp. 584–590
- XR Li, Y Bar-Shalom, Design of an interacting multiple model algorithm for air traffic control tracking. *Control Syst. Technol. IEEE Trans.* **1**(3), 186–194 (1993)
- XR Li, VP Jilkov, Survey of maneuvering target tracking. part i. dynamic models. *Aerosp. Electron. Syst. IEEE Trans.* **39**(4), 1333–1364 (2003)
- Y He, G Wang, X Guan, *Information Fusion Theory With Applications*. (Publishing House of Electronics Industry, Beijing, 2010)
- B Fan, YL Qin, JT Wang, HT Xiao, in *2011 International Workshop on Multi-Platform/Multi-Sensor Remote Sensing and Mapping*. 3d multiple maneuvering targets tracking in active and passive radar composite guidance, (2011), pp. 1–6
- Q He, J Hu, RS Blum, Y Wu, Generalized cramer rao bound for joint estimation of target position and velocity for active and passive radar networks. *IEEE Trans. Signal Process.* **64**(8), 2078–2089 (2016)
- YD Zhang, MG Amin, B Himed, Joint dod/doa estimation in mimo radar exploiting time-frequency signal representations. *J. Adv. Signal Process.* **2012**(1), 1–10 (2012)
- D Smith, S Singh, Approaches to multisensor data fusion in target tracking: a survey. *Knowle. Data Eng. IEEE Trans.* **18**(12), 1696–1710 (2006)
- Y Wei, J Qiu, HR Karimi, M Wang, H_{∞} model reduction for continuous-time markovian jump systems with incomplete statistics of mode information. *Int. J. Syst. Sci.* **45**(7), 1496–1507 (2014)

31. Y Wei, J Qiu, HR Karimi, M Wang, Filtering design for two-dimensional markovian jump systems with state-delays and deficient mode information. *Inf. Sci.* **269**(4), 316–331 (2014)
32. Y Wei, J Qiu, HR Karimi, M Wang, A new design of H_∞ filtering for continuous-time markovian jump systems with time-varying delay and partially accessible mode information. *Signal Process.* **93**(9), 2392–2407 (2013)
33. J Qiu, Y Wei, HR Karimi, New approach to delay-dependent H_∞ control for continuous-time markovian jump systems with time-varying delay and deficient transition descriptions. *J. Frankl. Inst.* **352**(1), 189–215 (2015)
34. L Li, SX Ding, J Qiu, Y Yang, Weighted fuzzy observer-based fault detection approach for discrete-time nonlinear systems via piecewise-fuzzy lyapunov functions. *IEEE Trans. Fuzzy Syst.* (99), 1–13 (2016)
35. L Li, SX Ding, J Qiu, Y Yang, Real-time fault detection approach for nonlinear systems and its asynchronous t-s fuzzy observer-based implementation. *Cybernet. IEEE Trans.* (99), 1–12 (2016)
36. J Qiu, S Ding, H Gao, S Yin, Fuzzy-model-based reliable static output feedback h-infinity control of nonlinear hyperbolic pde systems. *IEEE Trans. Fuzzy Syst.* **24**(2), 388–400 (2015)

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com
