Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A random finite set based joint probabilistic data association filter with non-homogeneous Markov chain*

Yun ZHU[†1,2], Shuang LIANG[3], Xiaojun WU[†‡1,2], Honghong YANG[1,2]

[1]*Key Laboratory of Modern Teaching Technology, Ministry of Education, Xi'an 710062, China*

[2]*School of Computer Science, Shaanxi Normal University, Xi'an 710119, China*

[3]*Academy of Advanced Interdisciplinary Research, Xidian University, Xi'an 710071, China*

[†]E-mail: yunzhu@snnu.edu.cn; xjwu@snnu.edu.cn

Received May 1, 2020; Revision accepted July 2, 2020; Crosschecked July 14, 2021

Abstract: We demonstrate a heuristic approach for optimizing the posterior density of the data association tracking algorithm via the random finite set (RFS) theory. Specifically, we propose an adjusted version of the joint probabilistic data association (JPDA) filter, known as the nearest-neighbor set JPDA (NNSJPDA). The target labels in all possible data association events are switched using a novel nearest-neighbor method based on the Kullback–Leibler divergence, with the goal of improving the accuracy of the marginalization. Next, the distribution of the target-label vector is considered. The transition matrix of the target-label vector can be obtained after the switching of the posterior density. This transition matrix varies with time, causing the propagation of the distribution of the target-label vector to follow a non-homogeneous Markov chain. We show that the chain is inherently doubly stochastic and deduce corresponding theorems. Through examples and simulations, the effectiveness of NNSJPDA is verified. The results can be easily generalized to other data association approaches under the same RFS framework.

Key words: Target tracking; Filtering theory; Random finite set theory; Bayes methods; Markov chain
https://doi.org/10.1631/FITEE.2000209 **CLC number:** TN713

1 Introduction

In multi-target tracking applications, multiple single-target Bayes trackers are generally used simultaneously to track multiple targets (Li et al., 2017). To achieve this, a data association algorithm is necessary for determining the correspondence between measurements and targets. Then, the tracking filter seeks to marginalize over the association events to calculate the marginal density of each track. A typical example of such algorithms is the joint probabilistic data association (JPDA) filter (Bar-Shalom and Tse, 1975; Fortmann et al., 1983). In situations characterized by high target densities, data association is often ambiguous. If false association lasts some time, tracks on closely spaced targets will tend to coalesce (Fitzgerald, 1985). To circumvent track coalescence, Fitzgerald (1990) proposed the exact nearest-neighbor JPDA (ENNJPDA) filter, where only the association event with the highest posterior probability is used for marginalization. The ENNJPDA filter can effectively address track coalescence, but it performs poorly, resulting in applications with false and missed detections. An alternative is the JPDA* filter, developed to conserve JPDA's resistance to false and missed detections while avoiding track coalescence (Bloem and Blom, 1995). When estimating the target states, these JPDA-based methods consider only the

‡ Corresponding author

* Project supported by the National Key Research and Development Program of China (No. 2017YFB1402102), the National Natural Science Foundation of China (Nos. 61907028 and 11872036), the Natural Science Foundation of Shaanxi Province, China (Nos. 2020JQ-423, 2019JQ-574, and 2019ZDLSF07-01), the Fundamental Research Funds for the Central Universities, China (No. GK201903103), and the China Postdoctoral Science Foundation (No. 2018M640950)

ORCID: Yun ZHU, https://orcid.org/0000-0001-7009-7627; Xiaojun WU, https://orcid.org/0000-0002-7779-553X

current density of the targets. An approach that considers the joint posterior of states over time will not have the track coalescence problem (Vo BN and Vo, 2019). However, the computation load of such a multi-scan solution can be large. Another method of avoiding ambiguous association is the use of a multi-hypothesis tracking (MHT) algorithm, which propagates all possible data association events over time (Reid, 1979). The correct association is always obtainable from sufficient association events. However, in the case of a large number of targets and measurements, MHT becomes computation demanding.

Recently, a lot of work has been devoted to random finite set (RFS) based multi-target tracking, with fruitful results. The finite set statistics (FISST) developed by Mahler (2007b) is among the most important results. Under the multi-target Bayes formulation, FISST efficiently avoids explicit association between measurements and targets. After prediction, the filter state is updated using a multi-target likelihood, which is a measure of the affinity between the predicted environment and the received measurement set. Some sequential Monte Carlo (SMC) implementations (Sidenbladh and Wirkander, 2003; Vo BN et al., 2005; Mahler, 2007b) of the filter have been proposed which have huge computational burden due to the complexity of the multi-target likelihood. To reduce computational complexity and enable practical application, Mahler proposed approximations like the probability hypothesis density (PHD) filter and the cardinalized probability hypothesis density (CPHD) filter (Mahler, 2003, 2007a). For the PHD and CPHD filters, implementations are possible using SMC methods (Sidenbladh, 2003; Zajic and Mahler, 2003; Vo BN et al., 2005) and Gaussian mixtures (Vo BN and Ma, 2006). Another approximation is the multi-Bernoulli filter (Vo BT et al., 2009; Li et al., 2020), which represents the multi-target posterior density using a multi-Bernoulli RFS. These filters reduce the computational complexity at the cost of losing the ability to easily model interactions and dependencies between targets (Reuter et al., 2013). Further, a closed-form solution of the multi-target Bayes filter has been proposed using the labeled RFS in several studies (Vo BT and Vo, 2013; Reuter et al., 2014; Vo BN et al., 2014; Beard et al., 2015). By combining the advantages of the RFS theory and data association methods, another kind of approach has been developed (Svensson D et al., 2011; Svensson L et al., 2011; Zhu et al., 2017, 2019). Collectively, these studies show that the tracking accuracy of the data association filter can be improved by switching the posterior densities in its RFS family. However, the computational complexity is usually high and the accuracy of the correspondence between labels and targets is not considered in these studies.

Assuming that the target number is known and that the targets are independent, we propose a heuristics approach for efficiently optimizing the posterior density of the data association filter while propagating the distribution of the target-label vectors. Since the JPDA filter is one of the most typical data association methods, we specifically consider this filter and develop an approach named nearest-neighbor set JPDA (NNSJPDA). The posterior density is optimized by switching the target labels in each data association event, aiming to improve the accuracy of marginalization. The target labels in each data association event are switched by a succinct nearest-neighbor method based on the Kullback–Leibler (KL) divergence (Kullback, 1968). After optimization, the transition matrix of the target-label vector is computed. Then, the propagation of the distribution for the target-label vector is modeled as a non-homogenous Markov chain using the transition matrix. The distribution of the target-label vector gives the user additional information about the target label, which is not available in conventional approaches. Moreover, some interesting properties and theorems of the non-homogenous Markov chain can be deduced. We will show that the chain is not only doubly stochastic but also ergodic, and that it always converges to the equilibrium state. The promising results can be generalized to study the target-label issue in other label-switching-based approaches under the same RFS framework, such as Svensson D et al. (2011), Svensson L et al. (2011), Williams (2015), Garcia-Fernandez (2016), and Zhu et al. (2017, 2019). The preliminary work is Liang et al. (2019). The current study provides a theoretical description of the optimization of posterior density and considers the target-label switching problem.

2 Problem formulation

To formulate the problem, we introduce the vector of ordered (labeled) target states at time k:

$$\boldsymbol{X}(k)=[(\boldsymbol{x}^1(k))^{\mathrm{T}},\ (\boldsymbol{x}^2(k))^{\mathrm{T}},\ \cdots,\ (\boldsymbol{x}^n(k))^{\mathrm{T}}]^{\mathrm{T}}, \quad (1)$$

where n is the number of targets, and $\boldsymbol{x}^i(k)=[x_k^i, \dot{x}_k^i, y_k^i, \dot{y}_k^i]^{\mathrm{T}}$ (i is the target label, and i=1, 2, …, n) is the state vector of a target with position $\boldsymbol{p}_k^i=[x_k^i, y_k^i]^{\mathrm{T}}$ and velocity $\boldsymbol{v}_k^i=[\dot{x}_k^i, \dot{y}_k^i]^{\mathrm{T}}$. Since an RFS is not ordered, the targets in Eq. (1) can be re-ordered without affecting the RFS. For example, when the target number n=2, the RFS family is expressed as

$$\{\boldsymbol{x}^1(k), \boldsymbol{x}^2(k)\}=\{\boldsymbol{a}^1, \boldsymbol{a}^2\}=\sum_{j=1}^{2}\boldsymbol{X}_j(k)=\sum_{j=1}^{2}\boldsymbol{\pi}_j\boldsymbol{a}, \quad (2)$$

where $\boldsymbol{a}=[\boldsymbol{a}^1, \boldsymbol{a}^2]$ is a point in the joint target state space and $\boldsymbol{\pi}_j$ is a permutation matrix. Since the dimensionality of $\boldsymbol{x}(k)$ is four, the two permutation matrices are as follows:

$$\boldsymbol{\pi}_1=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\otimes\boldsymbol{I}_4,\ \boldsymbol{\pi}_2=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\otimes\boldsymbol{I}_4. \quad (3)$$

The set of validated measurements collected at time k is denoted as

$$\boldsymbol{Z}(k)=\{\boldsymbol{z}_1(k), \boldsymbol{z}_2(k), \cdots, \boldsymbol{z}_{m_k}(k)\}, \quad (4)$$

where m_k denotes the number of measurements. The measurement set $\boldsymbol{Z}(k)$ comprises target-generated measurements and false measurements. Targets are detected with detection probability P_{d}. The false measurements are assumed to be Poisson distributed in number, with rate $r=\lambda|\mathrm{FoV}|$, where λ is the clutter intensity and |FoV| is the sensor's field of view. All false measurements distribute uniformly in the surveillance area.

3 Nearest-neighbor set joint probabilistic data association filter

A typical approach for the problem of tracking a fixed number of targets is the JPDA filter, the essence of which is the association of measurements and targets. At time k, the probability $P(\theta_h)$ of data association event θ_h (h=1, 2, …, N_{H}) is given by

$$P(\theta_h)=\bar{P}(\theta_h)\Big/\sum_{h=1}^{N_{\mathrm{H}}}\bar{P}(\theta_h), \quad (5)$$

$$\bar{P}(\theta_h)=\frac{\phi!}{V^{\phi}}\prod_{j=1}^{m_k}N_{tj}[\boldsymbol{z}_j(k)]^{f_j(\theta_h)}\cdot\prod_{t=1}^{n}(P_{\mathrm{d}}^t)^{\delta_t(\theta_h)}(1-P_{\mathrm{d}}^t)^{1-\delta_t(\theta_h)}, \quad (6)$$

$$N_{tj}[\boldsymbol{z}_j(k)]=N\,[\boldsymbol{z}_j(k), \boldsymbol{z}^t(k\,|\,k-1), \boldsymbol{S}^t(k)], \quad (7)$$

where ϕ is the number of false measurements, V is the volume of the valid space, $\boldsymbol{z}^t(k\,|\,k-1)$ is the predicted measurement of target t, $f_j(\theta_h)$ is the measurement indicator, and $\delta_t(\theta_h)$ is the target indicator. The target state and covariance at time k are updated as

$$\boldsymbol{x}^t(k\,|\,k)=\sum_{h=1}^{N_{\mathrm{H}}}P(\theta_h)\boldsymbol{x}_h^t(k\,|\,k), \quad (8)$$

$$\boldsymbol{P}^t(k\,|\,k)=\sum_{h=1}^{N_{\mathrm{H}}}P(\theta_h)\Big[\boldsymbol{P}_h^t(k\,|\,k)+\boldsymbol{x}_h^t(k\,|\,k)\boldsymbol{x}_h^{t'}(k\,|\,k)\Big]-\boldsymbol{x}^t(k\,|\,k)\boldsymbol{x}^{t'}(k\,|\,k), \quad (9)$$

where $\boldsymbol{x}_h^t(k\,|\,k)$ and $\boldsymbol{P}_h^t(k\,|\,k)$ are the updated state and covariance of target t in θ_h, respectively. Note that the standard JPDA filter updates each target by a weighted average of the measurements, whereas an alternative formulation is used here. For the proof of equivalence, refer to Liang et al. (2019).

In what follows, the posterior density $f(\boldsymbol{X})$ is modeled as a Gaussian mixture:

$$f(\boldsymbol{X})=\sum_{h=1}^{N_{\mathrm{H}}}P(\theta_h)f_h(\boldsymbol{X}), \quad (10)$$

where $f_h(\boldsymbol{X})=N(\boldsymbol{X}; \boldsymbol{x}_h, \boldsymbol{P}_h)$ is the Gaussian density of θ_h with mean $\boldsymbol{x}_h$ and covariance matrix $\boldsymbol{P}_h$. Here, $\boldsymbol{x}_h$ is the stacked vector of the updated target states in θ_h and $\boldsymbol{P}_h$ is the state covariance of $\boldsymbol{x}_h$. Assuming that the targets being tracked are independent of each other, $\boldsymbol{P}_h$ is denoted as

$$\boldsymbol{P}_h=\mathrm{diag}(\boldsymbol{P}_h^1, \boldsymbol{P}_h^2, \cdots, \boldsymbol{P}_h^n). \quad (11)$$

All covariance matrices are assumed to be strictly positive definite, so another description of Eq. (10) is

$$f(\boldsymbol{X}) = \sum_{h=1}^{N_{\mathrm{H}}} \frac{P(\theta_h)}{\sqrt{\det(2\pi \boldsymbol{P}_h)}} \exp\left[-\frac{1}{2}(\boldsymbol{X}-\boldsymbol{x}_h)^{\mathrm{T}} \boldsymbol{P}_h^{-1}(\boldsymbol{X}-\boldsymbol{x}_h)\right]. \tag{12}$$

Depending on the choice of $f(\boldsymbol{X})$, the Gaussian approximation can be more or less accurate. It is assumed that the degenerate Gaussian mixture with a single Gaussian component is given by

$$g(\boldsymbol{X}) = N(\boldsymbol{X}; \bar{\boldsymbol{X}}, \boldsymbol{R}) = \frac{1}{\sqrt{\det(2\pi \boldsymbol{R})}} \exp\left[-\frac{1}{2}(\boldsymbol{X}-\bar{\boldsymbol{X}})^{\mathrm{T}} \boldsymbol{R}^{-1}(\boldsymbol{X}-\bar{\boldsymbol{X}})\right], \tag{13}$$

where $\bar{\boldsymbol{X}}$ and $\boldsymbol{R}$ are computed as

$$\bar{\boldsymbol{X}} = E_{p(\boldsymbol{X})}\{\boldsymbol{X}\}, \quad \boldsymbol{R} = \mathrm{Cov}_{p(\boldsymbol{X})}\{\boldsymbol{X}\}. \tag{14}$$

Here $E\{\}$ and $\mathrm{Cov}\{\}$ represent the expected value and covariance, respectively.

3.1 Posterior density switching within the RFS family

The KL divergence has been widely used to measure the dissimilarity between two probability densities. In this study, it is used to measure the accuracy of the Gaussian approximation. If $g(\boldsymbol{X})$ and $f(\boldsymbol{X})$ are more similar, the KL divergence between them is smaller and the Gaussian approximation is more accurate, and vice versa. The KL divergence between $f(\boldsymbol{X})$ and $g(\boldsymbol{X})$ is defined as

$$d_{\mathrm{KL}}\left[f(\boldsymbol{X}) \,\|\, g(\boldsymbol{X})\right] = \int f(\boldsymbol{X}) \log \frac{f(\boldsymbol{X})}{g(\boldsymbol{X})} \delta \boldsymbol{X}. \tag{15}$$

By optimizing over $f(\boldsymbol{X})$, the KL divergence can be decreased. Ideally, $f(\boldsymbol{X})$ is optimized such that the KL divergence is minimized. Unfortunately, the KL divergence is not tractable, since $f(\boldsymbol{X})$ is expressed as a Gaussian mixture model. In general, the Monte Carlo (MC) sampling method is used to solve this problem, but it requires many samples and can be inefficient. To facilitate the computations, the KL divergence defined in Eq. (15) is approximated as

$$D_{\mathrm{KL}}\left[f(\boldsymbol{X}) \,\|\, g(\boldsymbol{X})\right] = \sum_{h=1}^{N_{\mathrm{H}}} P(\theta_h) d_{\mathrm{KL}}\left[f_h(\boldsymbol{X}) \,\|\, g(\boldsymbol{X})\right], \tag{16}$$

where $d_{\mathrm{KL}}\left[f_h(\boldsymbol{X}) \,\|\, g(\boldsymbol{X})\right]$ is the KL divergence between $f_h(\boldsymbol{X})$ and $g(\boldsymbol{X})$. The approximated KL divergence is an upper bound on the KL divergence, i.e., $D_{\mathrm{KL}}\left[f(\boldsymbol{X}) \,\|\, g(\boldsymbol{X})\right] \geq d_{\mathrm{KL}}\left[f(\boldsymbol{X}) \,\|\, g(\boldsymbol{X})\right]$. This is based on the convexity of the KL divergence (Kullback, 1968). Since a direct measure of the KL divergence would appear to be computation demanding, a compromise approach would be to use Eq. (16) as the cost function for posterior density optimization.

To make $g(\boldsymbol{X})$ approximate $f(\boldsymbol{X})$ more accurately, Svensson D et al. (2011) and Svensson L et al. (2011) showed that it is feasible to switch the ordered posterior density in the corresponding RFS family. Generally speaking, one can reorder the permutation of targets in each association event to decrease the dissimilarity between $f(\boldsymbol{X})$ and $g(\boldsymbol{X})$. In this study we develop a concise nearest-neighbor method to minimize the KL convergence (see Eq. (16)). In θ_h (h=1, 2, …, N_{H}), we enumerate all of the permutations of targets, and the permutation corresponding to the minimum KL divergence with $g(\boldsymbol{X})$ is selected. In mathematics,

$$\boldsymbol{\pi} = \arg\min_{\boldsymbol{\pi}} d_{\mathrm{KL}}\left[f_h^{\boldsymbol{\pi}}(\boldsymbol{X}) \,\|\, g(\boldsymbol{X})\right], \ \boldsymbol{\pi} \in \Pi_n, \tag{17}$$

where $f_h^{\boldsymbol{\pi}}(\boldsymbol{X})$ is the posterior density of θ_h under permutation $\boldsymbol{\pi}$, Π_n is the set of all possible permutations of targets, and $d_{\mathrm{KL}}[f_h^{\boldsymbol{\pi}}(\boldsymbol{X}) \,\|\, g(\boldsymbol{X})]$ is the KL divergence between $f_h^{\boldsymbol{\pi}}(\boldsymbol{X})$ and $g(\boldsymbol{X})$, which is computed as in Liang et al. (2019):

$$d_{\mathrm{KL}}[f_h^{\boldsymbol{\pi}}(\boldsymbol{X}) \,\|\, g(\boldsymbol{X})] = \frac{1}{2}\left[2\log\left(\det(\boldsymbol{R}_h^{\boldsymbol{\pi}})\right) - \log\left(\det(\boldsymbol{P}_h^{\boldsymbol{\pi}})\right) - \log\left(\det(\boldsymbol{R})\right)\right], \tag{18}$$

where

$$\boldsymbol{R}_h^{\boldsymbol{\pi}} = \boldsymbol{P}_h^{\boldsymbol{\pi}} + \boldsymbol{R} + (\boldsymbol{x}_h^{\boldsymbol{\pi}} - \bar{\boldsymbol{X}})(\boldsymbol{x}_h^{\boldsymbol{\pi}} - \bar{\boldsymbol{X}})^{\mathrm{T}}. \tag{19}$$

In this way, the KL divergence defined in Eq. (16) is decreased.

To further decrease the KL divergence, we employ an iterative strategy. Assume that the number of all iterations is denoted as L. Thus, the computational complexity of the posterior density optimization is $O((4n)^3 N_{\mathrm{H}} L)$. The detailed pseudocode is summarized in Algorithm 1.

Algorithm 1 Posterior density optimization of NNSJPDA

1. Initiate the parameters with i=0 and inputs $\boldsymbol{Z}(k)$, $\boldsymbol{X}(k-1)$, $\boldsymbol{P}(k-1)$
2. Formulate all N_{H} global data association events and compute their posterior probabilities using Eq. (5)
3. Update the target state $\boldsymbol{x}_h^t(k\,|\,k)$ and covariance matrix $\boldsymbol{P}_h^t(k\,|\,k)$ for each target t (t=1, 2, …, n) in each association event θ_h
4. Obtain the joint state vector and its corresponding covariance matrix using Eqs. (1) and (11), respectively
5. Approximate the posterior density with a single Gaussian $g_i(\boldsymbol{X})=N(\boldsymbol{X};\bar{\boldsymbol{X}}_i,\boldsymbol{R}_i)$ using Eq. (14)
6. Switch the index order in each hypothesis and set i=i+1
for data association event θ_h (h=1, 2, …, N_{H}) **do**
  for each permutation of targets $\boldsymbol{\pi}\in\Pi_n$ **do**
    Compute the KL divergence $d_{\mathrm{KL}}[f_h^{\pi}(\boldsymbol{X})\,\|\,g(\boldsymbol{X})]$
  end for
  Select the permutation with the minimum KL divergence
end for
7. Approximate the new posterior density with a new Gaussian density $g_i(\boldsymbol{X})=N(\boldsymbol{X};\bar{\boldsymbol{X}}_i,\boldsymbol{R}_i)$
8. Compute $D_{\mathrm{KL}}^i\left[f_i(\boldsymbol{X})\,\|\,g_i(\boldsymbol{X})\right]$ using Eq. (16). If $D_{\mathrm{KL}}^i\left[f_i(\boldsymbol{X})\,\|\,g_i(\boldsymbol{X})\right]\approx D_{\mathrm{KL}}^{i-1}\left[f_{i-1}(\boldsymbol{X})\,\|\,g_{i-1}(\boldsymbol{X})\right]$, stop; otherwise, go back to step 6

The following example illustrates the posterior density optimization. It is assumed that there are two Gaussian distributed one-dimensional targets which generate two data association events θ_1 and θ_2. The initial posterior density $f^0(\boldsymbol{X})$ is denoted as

$$\begin{cases}P(\theta_1)=0.5,\\ \theta_1\sim N\left(\boldsymbol{X};\begin{bmatrix}3\\1\end{bmatrix},\begin{bmatrix}1&0\\0&1\end{bmatrix}\right),\end{cases}\quad\begin{cases}P(\theta_2)=0.5,\\ \theta_2\sim N\left(\boldsymbol{X};\begin{bmatrix}0\\4\end{bmatrix},\begin{bmatrix}1&0\\0&1\end{bmatrix}\right).\end{cases}\tag{20}$$

The single Gaussian approximation is

$$g^0(\boldsymbol{X})=N\left(\boldsymbol{X};\begin{bmatrix}1.5\\2.5\end{bmatrix},\begin{bmatrix}3.25&0\\0&3.25\end{bmatrix}\right).\tag{21}$$

The approximate KL divergence between $f^0(\boldsymbol{X})$ and $g^0(\boldsymbol{X})$ is 0.5119. Then, we perform the iterative optimization to obtain a new $f(\boldsymbol{X})$ that more closely represents $g(\boldsymbol{X})$. The algorithm converges quickly such that only two iterations are performed. The resulting posterior density $f^2(\boldsymbol{X})$ is

$$\begin{cases}P(\theta_1)=0.5,\\ \theta_1\sim N\left(\boldsymbol{X};\begin{bmatrix}1\\3\end{bmatrix},\begin{bmatrix}1&0\\0&1\end{bmatrix}\right),\end{cases}\quad\begin{cases}P(\theta_2)=0.5,\\ \theta_2\sim N\left(\boldsymbol{X};\begin{bmatrix}0\\4\end{bmatrix},\begin{bmatrix}1&0\\0&1\end{bmatrix}\right).\end{cases}\tag{22}$$

Its single Gaussian approximation is

$$g^2(\boldsymbol{X})=N\left(\boldsymbol{X};\begin{bmatrix}0.5\\3.5\end{bmatrix},\begin{bmatrix}1.25&0\\0&1.25\end{bmatrix}\right).\tag{23}$$

The approximate KL divergence between $f^2(\boldsymbol{X})$ and $g^2(\boldsymbol{X})$ is decreased to 0.0969. We plot $f^0(\boldsymbol{X})$ and $g^0(\boldsymbol{X})$ in Figs. 1a and 1b, respectively. The resulting $f^2(\boldsymbol{X})$ and $g^2(\boldsymbol{X})$ are plotted in Figs. 1c and 1d, respectively. It is obvious that the initial densities are very different from each other, and that the similarity between $f^2(\boldsymbol{X})$ and $g^2(\boldsymbol{X})$ becomes much higher than it was initially.

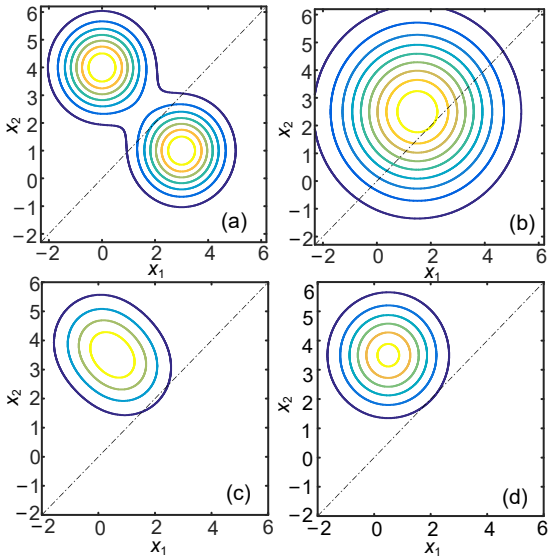

Fig. 1 Posterior density and its single Gaussian density: (a) posterior density of iteration 0; (b) single Gaussian of iteration 0; (c) posterior density of iteration 2; (d) single Gaussian of iteration 2

3.2 Non-homogeneous Markov chain for label propagation

In the posterior density optimization procedure, the output is the best permutation of targets in each association event. Each event is associated with a posterior probability, and this probability also carries the information about the permutation of target labels. We define the probability of a target label at time k as

the probability of a label vector $\boldsymbol{L}_j$, i.e.,

$$P_k^L(\boldsymbol{L}_j)\triangleq\Pr\{\boldsymbol{L}_k=\boldsymbol{\pi}_j\boldsymbol{L}_0\,|\,\boldsymbol{Z}_k\},\ j=1,2,\cdots,n!, \quad (24)$$

where $\boldsymbol{L}_0\triangleq[1,2,\cdots,n]^{\mathrm{T}}$ is the initial label vector, and $n!$ is the total number of all the label vectors.

For example, when there are two targets and n=2, there will exist two permutations of target labels, that is, $\boldsymbol{L}_1$=[1, 2]$^{\mathrm{T}}$ and $\boldsymbol{L}_2$=[2, 1]$^{\mathrm{T}}$. We use

$$P_k^L(\boldsymbol{L}_1)=\Pr\left\{\boldsymbol{L}_k=\begin{bmatrix}1\\2\end{bmatrix}\middle\|\,\boldsymbol{Z}_k\right\} \quad (25)$$

and

$$P_k^L(\boldsymbol{L}_2)=\Pr\left\{\boldsymbol{L}_k=\begin{bmatrix}2\\1\end{bmatrix}\middle\|\,\boldsymbol{Z}_k\right\} \quad (26)$$

to express the probabilities of the target labels. The propagation of these probabilities can be modeled as a Markov chain (Svensson D et al., 2011; Svensson L et al., 2011), for two reasons: (1) the outcome of each time is one of the discrete states; (2) the outcome of the present time depends only on the past states, and not on any future states. The probabilities of labels can be computed as follows:

$$P_k^L(\boldsymbol{L}_1)=t_{k|k-1}^{11}P_{k-1}^L(\boldsymbol{L}_1)+t_{k|k-1}^{21}P_{k-1}^L(\boldsymbol{L}_2), \quad (27)$$

$$P_k^L(\boldsymbol{L}_2)=t_{k|k-1}^{22}P_{k-1}^L(\boldsymbol{L}_2)+t_{k|k-1}^{12}P_{k-1}^L(\boldsymbol{L}_1), \quad (28)$$

where $0\le t_{k|k-1}^{ij}\le 1$ is the transition probability from label vector $\boldsymbol{L}_i$ to $\boldsymbol{L}_j$.

The recursion of $P_k^L(\boldsymbol{L}_j)$ is then generalized for the n-target case. Assuming that all transition probabilities are known, the recursion of $P_k^L(\boldsymbol{L}_j)$ is

$$P_k^L(\boldsymbol{L}_j)=\sum_{i=1}^{n!}t_{k|k-1}^{ij}P_{k-1}^L(\boldsymbol{L}_i). \quad (29)$$

The vector $\boldsymbol{P}_k^L=[P_k^L(\boldsymbol{L}_1),P_k^L(\boldsymbol{L}_2),\ldots,P_k^L(\boldsymbol{L}_{n!})]^{\mathrm{T}}$ can be updated as a matrix product:

$$\boldsymbol{P}_k^L=\boldsymbol{T}_{k|k-1}\boldsymbol{P}_{k-1}^L, \quad (30)$$

where

$$\boldsymbol{T}_{k|k-1}=\begin{bmatrix}t_{k|k-1}^{11} & t_{k|k-1}^{12} & \cdots & t_{k|k-1}^{1n!}\\ t_{k|k-1}^{21} & t_{k|k-1}^{22} & \cdots & t_{k|k-1}^{2n!}\\ \vdots & \vdots & & \vdots\\ t_{k|k-1}^{n!1} & t_{k|k-1}^{n!2} & \cdots & t_{k|k-1}^{n!n!}\end{bmatrix} \quad (31)$$

is an $n!\times n!$ transition matrix with element (i, j) being $t_{k|k-1}^{ij}$. The probability vector $\boldsymbol{P}_k^L$ can be propagated over time using Eq. (30). With the propagation of $\boldsymbol{P}_k^L$, the approach has the ability to preserve uncertainties of the target labels.

The transition probability $t_{k|k-1}^{ij}$ is related to the probability of each association event. In event θ_h, we use $\delta_{k|k-1}^{ij}(\theta_h)$ as the permutation indicator (if label vector $\boldsymbol{L}_{k-1}^i$ is switched to $\boldsymbol{L}_k^j$ in θ_h, the indicator $\delta_{k|k-1}^{ij}(\theta_h)$=1; otherwise, $\delta_{k|k-1}^{ij}(\theta_h)$=0). Then, the transition probability $t_{k-1|k}^{ij}$ is expressed as

$$t_{k|k-1}^{ij}=\sum_{h=1}^{N_{\mathrm{H}}}P(\theta_h)\delta_{k|k-1}^{ij}(\theta_h). \quad (32)$$

The transition matrix varies with time, so the Markov chain is non-homogeneous. Compared to the homogeneous case, the non-homogeneous Markov chain has received less attention. We will prove that with the increase of time, the probabilities of all possible label vectors will be equal. To clarify the problem, consider two definitions regarding the Markov chain below:

Definition 1 A discrete Markov chain $\{x_k\}$ (k=1, 2, …) with state space S={1, 2, …} is said to be stationary or homogeneous if the probability of going from one state to another is independent of the time at which the step is being made. That is, for all states, we have

$$\Pr(x_n=i_n\,|\,x_{n-1}=i_{n-1})=\Pr(x_{n+k}=i_{n+k}\,|\,x_{n+k-1}=i_{n+k-1}) \quad (33)$$

for k=−(n−1), −(n−2), …, −1, 0, 1, 2, …. The chain is said to be non-stationary or non-homogeneous if the condition for stationarity fails.

Definition 2 A vector $\boldsymbol{v}\in\mathbb{R}^n$ is stochastic if $\boldsymbol{v}\ge 0$ and $\sum_{i=1}^n v_i=1$. A matrix $\boldsymbol{A}$ is stochastic when all its rows are stochastic, and it is further doubly stochastic

when both $\boldsymbol{A}$ and $\boldsymbol{A}^{\mathrm{T}}$ are stochastic.

With the above definitions, we will show some properties and theorems of the proposed non-homogeneous Markov process. First, note that although the transition matrix of the chain is time-varying, the transition matrix is always doubly stochastic.

Property 1 At time k, if the $(i,j)^{\text{th}}$ element $t_{k|k-1}^{ij}$ of transition matrix $\boldsymbol{T}_{k|k-1}$ is modeled as in Eq. (32), the transition matrix is doubly stochastic.

Proof For the i^{th} ($1\leq i\leq n!$) row of the transition matrix, we have

$$\sum_{j=1}^{n!} t_{k|k-1}^{ij} = \sum_{j=1}^{n!}\sum_{h=1}^{N_{\mathrm{H}}} P(\theta_h)\delta_{k|k-1}^{ij}(\theta_h)=1. \tag{34}$$

In fact, it is also a proof that the proposed transition matrix satisfies the feature of the Markov process. That is, the sum of the entries in any row must be 1, since the entries in the row give the probability of changing from the state at the previous time step to one of the current states. For the j^{th} ($1\leq j\leq n!$) column of the transition matrix, we have

$$\sum_{j=1}^{n!} t_{k|k-1}^{ij} = \sum_{j=1}^{n!}\sum_{h=1}^{N_{\mathrm{H}}} P(\theta_h)\delta_{k|k-1}^{ij}(\theta_h)=1. \tag{35}$$

From Eqs. (34) and (35), it is proved that the transition matrix is doubly stochastic.

With the increase of time, the distribution $\boldsymbol{P}_k^L$ of label vectors will be propagated through the transition probability matrix $\boldsymbol{T}_{k|k-1}$. At time $K>k$, the distribution $\boldsymbol{P}_k^L$ is computed as

$$\boldsymbol{P}_K^L = \boldsymbol{T}^{k,K}\boldsymbol{P}_k^L, \tag{36}$$

where $\boldsymbol{T}^{k,K}=\boldsymbol{T}_{k+1|k}\boldsymbol{T}_{k+2|k+1}\cdots\boldsymbol{T}_{K|K-1}$ is the product of the transition matrices used. Since each transition matrix is doubly stochastic, the product of these transition matrices is also doubly stochastic.

Theorem 1 Let S denote the set of doubly stochastic matrices. The product of matrices in any subset $C\subset S$ is also doubly stochastic.

Proof If we can prove that the product of two random doubly stochastic matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ is still doubly stochastic, then Theorem 1 is proved. Let

$$\boldsymbol{A}=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix},\ \boldsymbol{B}=\begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1N} \\ b_{21} & b_{22} & \cdots & b_{2N} \\ \vdots & \vdots & & \vdots \\ b_{N1} & b_{N2} & \cdots & b_{NN} \end{bmatrix}. \tag{37}$$

For matrix $\boldsymbol{A}$, $\sum_{j=1}^{N} a_{ij}=1$ with $1\leq i\leq N$ and $\sum_{i=1}^{N} a_{ij}=1$ with $1\leq j\leq N$; for matrix $\boldsymbol{B}$, $\sum_{j=1}^{N} b_{ij}=1$ with $1\leq i\leq N$ and $\sum_{i=1}^{N} b_{ij}=1$ with $1\leq j\leq N$. For the product of $\boldsymbol{A}$ and $\boldsymbol{B}$,

$$\boldsymbol{A}\times\boldsymbol{B}=\begin{bmatrix} \sum_{i=1}^{N} a_{1i}b_{i1} & \sum_{i=1}^{N} a_{1i}b_{i2} & \cdots & \sum_{i=1}^{N} a_{1i}b_{iN} \\ \sum_{i=1}^{N} a_{2i}b_{i1} & \sum_{i=1}^{N} a_{2i}b_{i2} & \cdots & \sum_{i=1}^{n!} a_{2i}b_{iN} \\ \vdots & \vdots & & \vdots \\ \sum_{i=1}^{N} a_{Ni}b_{i1} & \sum_{i=1}^{N} a_{Ni}b_{i2} & \cdots & \sum_{i=1}^{N} a_{Ni}b_{iN} \end{bmatrix}, \tag{38}$$

the sum of entries in the k^{th} ($1\leq k\leq N$) row of $\boldsymbol{A}\times\boldsymbol{B}$ is

$$\sum_{j=1}^{N}\sum_{i=1}^{N} a_{ki}b_{ij} = \sum_{i=1}^{N} a_{ki}\sum_{j=1}^{N} b_{ij} = \sum_{i=1}^{N} a_{ki} = 1, \tag{39}$$

and the sum of entries in the l^{th} ($1\leq l\leq N$) column of $\boldsymbol{A}\times\boldsymbol{B}$ is

$$\sum_{j=1}^{N}\sum_{i=1}^{N} a_{ji}b_{il} = \sum_{i=1}^{N} b_{il}\sum_{j=1}^{N} a_{ji} = \sum_{i=1}^{N} b_{il} = 1. \tag{40}$$

It can be seen that $\boldsymbol{A}\times\boldsymbol{B}$ is also a doubly stochastic matrix. It follows that any product of doubly stochastic matrices is also doubly stochastic.

Property 2 At time k, if the $(i,j)^{\text{th}}$ element $t_{k-1|k}^{ij}$ of the transition matrix $\boldsymbol{T}_{k|k-1}$ is modeled as in Eq. (32), the non-homogeneous doubly stochastic Markov chain is ergodic.

Proof The Birkhoff–von Neumann theorem (Horn and Johnson, 1985) establishes that any doubly stochastic matrix can be written as a convex combination of the permutation matrices. Therefore, any doubly

stochastic chain is decomposable. For the proposed chain, it is decomposable with a trivial permutation component since entries on the diagonal are equal. In this case, the infinite flow property is equivalent to the absolute infinite flow property. Touri and Nedić (2011) proved that a doubly stochastic chain is ergodic if it has the property of absolute infinite flow. Thus, the proposed doubly stochastic non-homogeneous Markov chain is ergodic. This property will make the Markov chain converge. Not only does the Markov chain have the equilibrium vector, but the product of the transition matrices also converges to a unique matrix.

Theorem 2 For the ergodic doubly stochastic non-homogeneous Markov chain, we denote $\{\boldsymbol{A}_k|A_k(i,j)=a_k(i,j),\ k\geq 1,\ 1\leq i\leq N,\ 1\leq j\leq N\}$ as the set of transition matrices. There is a unique $N\times N$ matrix $\mathbf{1}$ with each entry equaling 1 such that

$$\lim_{K\to\infty}\left\|\boldsymbol{A}^{k,K}-\frac{1}{N}\mathbf{1}\right\|=0, \tag{41}$$

where $\boldsymbol{A}^{k,K}=\boldsymbol{A}_{k+1}\boldsymbol{A}_{k+2}\cdots\boldsymbol{A}_K$. For any initial distribution of the Markov chain, the output distribution is a fixed vector $(1/N)\boldsymbol{e}$, where $\boldsymbol{e}$ denotes the $N\times 1$ vector with all entries equal to one.

Proof Since the Markov chain is ergodic, it follows that $\lim_{K\to\infty}\left\|\boldsymbol{A}^{k,K}-\boldsymbol{Q}\right\|=0$, where $\boldsymbol{Q}$ is a row-constant matrix. Referring to Theorem 1, the product of doubly stochastic matrices $\boldsymbol{A}^{k,K}$ is still doubly stochastic, and hence $\boldsymbol{Q}$ must be doubly stochastic. For a row-constant matrix $\boldsymbol{Q}$, if it is doubly stochastic, then $\boldsymbol{Q}=(1/N)\mathbf{1}$. Whenever a stochastic vector is used to multiply the matrix $(1/N)\mathbf{1}$, the output will always be $(1/N)\boldsymbol{e}$.

4 Simulations

In this section, the NNSJPDA filter is evaluated in two scenarios. The first scenario is used to evaluate the performance of posterior density optimization, and the second is used to observe the propagation of the target-label vectors. Both scenarios under consideration involve the tracking of two targets. Their trajectories are shown in Fig. 2, in which the targets approach each other, then move in parallel at a close distance, and finally are separated from each other (Svensson D et al., 2011; Svensson L et al., 2011; Panakkal and Velmurugan, 2013; Jing et al., 2015; Zhu et al., 2017, 2019). The target dynamics is modeled by a linear Gaussian model:

$$\boldsymbol{x}(k+1)=\boldsymbol{F}\boldsymbol{x}(k)+\boldsymbol{u}(k), \tag{42}$$

where $\boldsymbol{F}$ is the transition matrix and $\boldsymbol{u}(k)\sim N(\mathbf{0},\boldsymbol{Q})$ is the zero-mean white Gaussian process noise with covariance matrix $\boldsymbol{Q}$. We adopt

$$\boldsymbol{F}=\boldsymbol{I}_2\otimes\begin{bmatrix}1 & T_s\\ 0 & 1\end{bmatrix},\ \boldsymbol{Q}=\boldsymbol{I}_2\otimes p\begin{bmatrix}T_s^3/3 & T_s^2/2\\ T_s^2/2 & T_s\end{bmatrix}, \tag{43}$$

where T_s is the sampling interval and p is a tuning parameter. For simplicity, it is assumed that the target position can be observed. The relationship between a target and its measurement is given by the measurement model:

$$\boldsymbol{z}(k)=\boldsymbol{H}\boldsymbol{x}(k)+\boldsymbol{w}(k), \tag{44}$$

where

$$\boldsymbol{H}=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix},\ \boldsymbol{w}(k)\sim N(\mathbf{0},\boldsymbol{R}). \tag{45}$$

The covariance matrix $\boldsymbol{R}$ for the Gaussian measurement noise $\boldsymbol{w}(k)$ is

$$\boldsymbol{R}=\begin{bmatrix}\sigma_x^2 & 0\\ 0 & \sigma_y^2\end{bmatrix}, \tag{46}$$

where σ_x^2 and σ_y^2 are noise deviations in the x and y dimensions, respectively. The clutter measurements are assumed to be Poisson distributed with intensity λ over the region |FoV|. It is noteworthy that although we assume linear models in this study, the proposed approach can be easily extended to handle nonlinear Gaussian dynamic and measurement models.

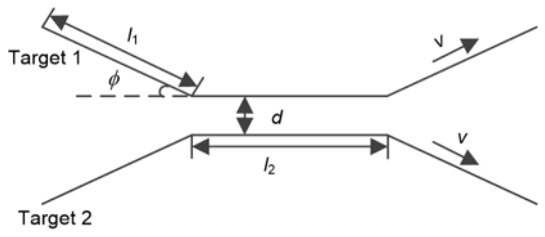

Fig. 2 Illustration of the scenario

In Fig. 2, the fixed values used in the simulations are $\phi=\pi/6$, $l_1=10$ m, and $l_2=10$ m. Targets exist for time indices $k=1, 2, \ldots, 31$ with speed $v=1$ m/s. The sampling interval $T_s=1$ s. For the clutter measurement, $\lambda=0.01$ m^{-2} and $|\text{FoV}|=1.4\times10^3$ m^2. All experiments are tested in Matlab R2010a and implemented on a computer with a 3.40 GHz processor.

4.1 Scenario 1

In the first scenario, the parameters used in the evaluations are $\sigma_x=\sigma_y=0.2$ m, $P_d=0.9$, $p=0.3$, and $d=0.5$ m. The target position estimates for a single run of the NNSJPDA and JPDA filters are illustrated in Figs. 3a and 3b, respectively.

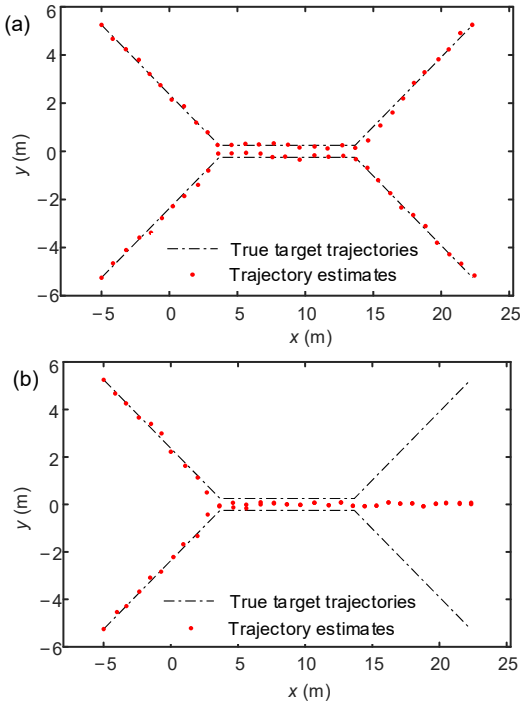

Fig. 3 True and estimated target trajectories: (a) output of NNSJPDA; (b) output of JPDA

It is evident from Fig. 3a that track coalescence does not occur in the NNSJPDA filter when targets are closely spaced, whereas this is not the case with the JPDA filter, as shown in Fig. 3b. The estimation accuracy of the algorithms is measured using the optimal sub-pattern assignment (OSPA) error distance. As the number of targets is assumed known, we present only the OSPA distance for a known number of targets. Let $\hat{X}=\{\hat{\boldsymbol{x}}_1,\hat{\boldsymbol{x}}_2,\cdots,\hat{\boldsymbol{x}}_n\}$ be the set of target state estimates and $X=\{x_1, x_2, \ldots, x_n\}$ the set of true target states, both with n elements. The OSPA distance between $\hat{X}$ and X is defined as in Schuhmacher et al. (2008):

$$d_{p,c}^{\text{OSPA}}(\hat{X},X)=\left(\frac{1}{n}\left(\min_{\boldsymbol{\pi}\in\Pi_n}\sum_{i=1}^{n}d^{(c)}(\boldsymbol{x}^{(i)},\hat{\boldsymbol{x}}^{\boldsymbol{\pi}(i)})^p\right)\right)^{1/p}, \quad (47)$$

where $d^{(c)}(\boldsymbol{x},\hat{\boldsymbol{x}})\triangleq\min(c,d(\boldsymbol{x},\hat{\boldsymbol{x}}))$ is the distance between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$, $c>0$ is the cut-off parameter, $p\geq1$ is a unitless real number, $\hat{\boldsymbol{x}}^{\boldsymbol{\pi}(i)}$ is the state vector corresponding to the i^{th} permutation, and Π_n is the set of all possible permutations of $\hat{X}$. In the following, we adopt the parameters of the OSPA metric as $p=1$, $c=0.4$ m.

To evaluate the tracking performance of the NNSJPDA filter, it is compared with the JPDA, ENNJPDA, SJPDA, and KLSJPDA (Svensson D et al., 2011; Svensson L et al., 2011) filters using the averaged OSPA (AOSPA) error. At each time step, the implementation of the SJPDA filter retains at most eight hypotheses with the largest probabilities, to reduce the computation burden. For the KLSJPDA filter, 100 samples are used for all iterations. The AOSPA performances over 500 MC trials are shown in Fig. 4. It can be observed that the performances of all filters vary with the change of the detection probability. For $P_d=1$, the AOSPA performances of all anti-coalescing methods are comparable with each other. However, the AOSPA performances of SJPDA, KLSJPDA, and NNSJPDA are significantly better than those of other algorithms when $P_d=0.75$. This is because the SJPDA, KLSJPDA, and NNSJPDA filters are designed under the RFS framework to minimize the AOSPA measure at the cost of losing target identity.

To study the performances of the SJPDA, KLSJPDA, and NNSJPDA filters in depth, we provide some results showing the relationship between P_d and the AOSPA error at a certain time for these four algorithms (Table 1). The results of the JPDA filter are used as the reference. Table 1 shows that the AOSPA performances of the KLSJPDA and NNSJPDA filters are better than that of the SJPDA filter. The KLSJPDA filter is derived in the optimal KL sense, but it performs only slightly better than the NNSJPDA filter.

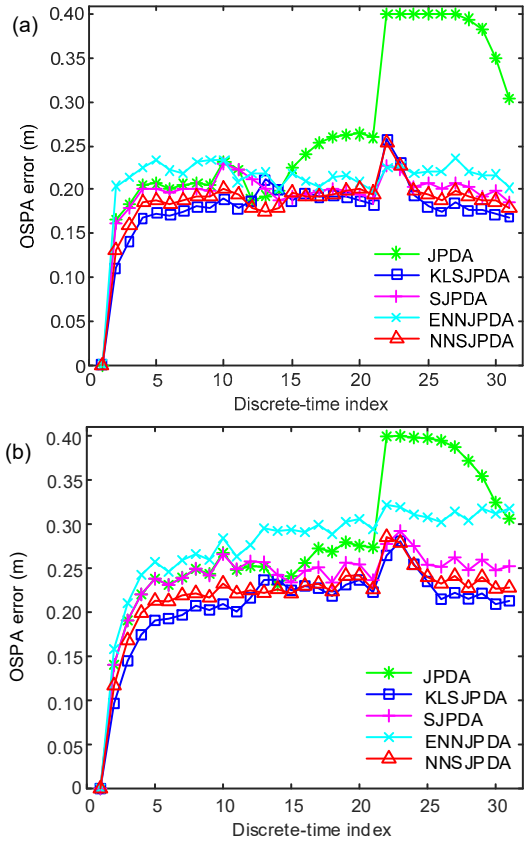

Fig. 4 Evaluation of the JPDA, ENNJPDA, SJPDA, KLSJPDA, and NNSJPDA filters for different detection probabilities: (a) P_d=1; (b) P_d=0.75

Table 1 Averaged OSPA error for a single run

Detection probability	Averaged OSPA error (cm)			
	JPDA	SJPDA	KLSJPDA	NNSJPDA
1.0	39.7	20.9	19.1	20.1
0.9	38.9	23.6	20.0	20.9
0.8	36.3	25.1	21.2	22.3

The track loss probability is also a generally used evaluation criterion of tracking algorithms and is considered in Table 2. A track is considered lost if the state covariance matrix elements corresponding to the x or y dimension surpass 2 m. Table 2 indicates that the track loss probabilities of the SJPDA, KLSJPDA, and NNSJPDA filters are dramatically reduced when compared with that of the JPDA filter, due to their avoidance of track coalescence. Going a step further, the track losses of the KLSJPDA and NNSJPDA filters occur less frequently than that of the SJPDA filter when P_d<1, because they can give better target locations.

Table 2 Track loss probabilities with different detection probabilities

Detection probability	Track loss probability			
	JPDA	SJPDA	KLSJPDA	NNSJPDA
1.0	0.22	0	0	0
0.9	0.27	0.02	0.01	0.01
0.8	0.32	0.05	0.01	0.02

The averaged computing time for the JPDA, SJPDA, KLSJPDA, and NNSJPDA filters against the averaged number of clutter points per scan varying from r=10 to 100 are shown in Fig. 5. There is an overall increase of computing time with the increase of r. This is because there are many false measurements which fall into the track gate when r is high and all the data association based methods can be time-consuming. However, the KLSJPDA filter needs much more computing time than the JPDA, SJPDA, and NNSJPDA filters. Note that only 100 samples are used during the iterative procedure for the KLSJPDA filter. Using more samples can improve the accuracy of the posterior density, which improves the performance of the KLSJPDA filter to a certain accuracy, but the computing time will further increase. The JPDA filter is about 2–3 times faster than the SJPDA filter, because the SJPDA filter establishes a goal function to measure the multimodality of the density. To minimize the goal function, one needs to search for all of the combinations of indexations of all hypotheses, which takes a long time. Compared with the SJPDA and KLSJPDA filters, as expected, the NNSJPDA filter clearly reduces the computation burden. Moreover, the computing time of the NNSJPDA filter is nearly equivalent to that of JPDA in the operational environment of these simulations. Therefore, the above simulations show that the estimation accuracy of the NNSJPDA filter is comparable to that of the optimal KLSJPDA filter. In terms of computing time, the proposed approach is significantly more efficient.

4.2 Scenario 2

In this subsection, simulations are conducted to demonstrate that the NNSJPDA filter has the potential to propagate the distribution of the target-label vectors. The NNSJPDA filter with the target label is evaluated using the scenario illustrated in Fig. 2, with p=0.08, P_d=0.9, and σ_x=σ_y=0.1 m. Three different

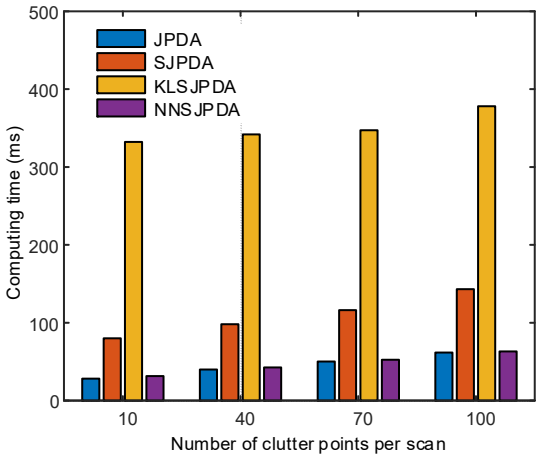

Fig. 5 Averaged computing time against the averaged number of clutter points per scan

separation distances d=1.5, 1, and 0.5 m are considered. The output of the JPDA filter is used as a reference. In Fig. 6, the position estimates for a single run of the JPDA and NNSJPDA filters with the target label are shown in the left and middle panels, respectively, and the probabilities of label vectors are shown in the right label.

In Fig. 6a, the probability that tracks follow the original targets is close to one throughout the scenario, because the targets move far away and switches of target labels seldom occur. In Fig. 6b, the probability of the original label vector slightly decreases to 0.5 after the two targets move close to each other. In Fig. 6c, the probability of the original label vector quickly drops to 0.5, which indicates that it is

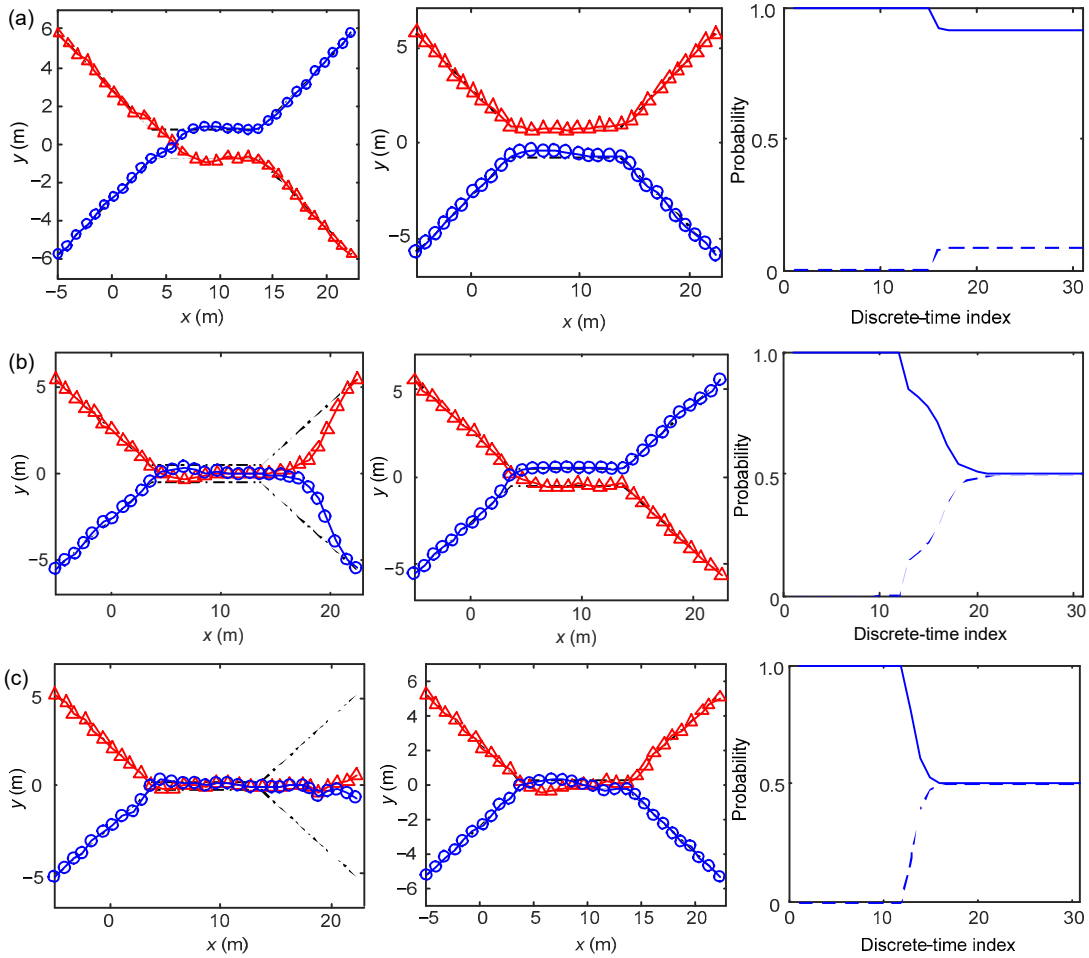

Fig. 6 Example output for d=1.5 m (a), d=1 m (b), and d=0.5 m (c)

The position estimates for a single run of the JPDA and NNSJPDA filters with the target label are shown in the left and middle panels, respectively, and the probabilities of label vectors are shown in the right panel. In the left and middle panels, the true target positions are indicated by dotted lines and the estimated target positions by "-○-" and "-△-." In the right panel, the solid line represents the probability that tracks are associated with the original targets and the dashed line denotes the probability of the opposite case

impossible to associate the tracks with the correct targets after they move close together. These results are in agreement with the theoretical analysis. In a word, the NNSJPDA filter has the ability to propagate the distribution of the label vectors, which indicates the uncertainty in track-to-target assignment. However, the JPDA filter does not provide such information, as it always assumes that the tracks are following the right targets even when false association occurs.

5 Conclusions

In this paper, we present a novel filter named NNSJPDA, which combines the RFS theory with the JPDA filter. To accurately approximate the posterior density of the NNSJPDA filter with a single Gaussian density, the posterior density is optimized by switching within its RFS family. Specifically, posterior density optimization is accomplished by switching the posterior densities of all possible data association events in parallel, aiming to minimize the KL divergence between the posterior density and the single Gaussian density. It is further shown that the information of the target label vectors can be preserved during the posterior density switching procedure. This information is used to propagate the distribution of the target label vectors, and the propagation is modeled as a non-homogeneous Markov chain. The chain is doubly stochastic and ergodic, and always converges to the equilibrium state whatever the initial distribution is. The simulation results show that the NNSJPDA filter can effectively avoid the well-known track-coalescence problem and estimate the target positions well. In terms of computation, the computing time of the NNSJPDA filter is comparable to that of the JPDA filter, because optimization of the posterior density needs less computational effort and converges quickly. One limitation of the NNSJPDA filter is that the target number must be given a priori. Thus, a future direction would be to adapt the proposed approach to uncertainty in the number of targets.

Contributors

Yun ZHU designed the research. Yun ZHU and Shuang LIANG drafted the manuscript. Xiaojun WU and Honghong YANG helped organize the manuscript and finalized the paper.

Compliance with ethics guidelines

Yun ZHU, Shuang LIANG, Xiaojun WU, and Honghong YANG declare that they have no conflict of interest.

References

Bar-Shalom Y, Tse E, 1975. Tracking in a cluttered environment with probabilistic data association. *Automatica*, 11(5):451-460. https://doi.org/10.1016/0005-1098(75)90021-7

Beard M, Reuter S, Granström K, et al., 2015. A generalised labelled multi-Bernoulli filter for extended multi-target tracking. Proc 18th Int Conf on Information Fusion, p.991-998.

Bloem EA, Blom HAP, 1995. Joint probabilistic data association methods avoiding track coalescence. Proc 34th IEEE Conf on Decision and Control, p. 2752-2757. https://doi.org/10.1109/CDC.1995.478532

Fitzgerald RJ, 1985. Track biases and coalescence with probabilistic data association. *IEEE Trans Aerosp Electron Syst*, AES-21(6):822-825. https://doi.org/10.1109/TAES.1985.310670

Fitzgerald RJ, 1990. Development of practical PDA logic for multitarget tracking by microprocessor. Proc American Control Conf, p.1-23. https://doi.org/10.23919/ACC.1986.4789059

Fortmann T, Bar-Shalom Y, Scheffe M, 1983. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE J Ocean Eng*, 8(3):173-183. https://doi.org/10.1109/JOE.1983.1145560

Garcia-Fernandez AF, 2016. Track-before-detect labeled multi-Bernoulli particle filter with label switching. *IEEE Trans Aerosp Electron Syst*, 52(5):2123-2138. https://doi.org/10.1109/TAES.2016.150343

Horn RA, Johnson CR, 1985. Matrix Analysis. Cambridge University Press, New York, USA.

Jing PL, Xu SY, Li X, et al., 2015. Coalescence-avoiding joint probabilistic data association based on bias removal. *EURASIP J Adv Signal Process*, 2015(1):24. https://doi.org/10.1186/s13634-015-0205-2

Kullback S, 1968. Information Theory and Statistics. Dover, New York, USA.

Li TC, Su JY, Liu W, et al., 2017. Approximate Gaussian conjugacy: parametric recursive filtering under nonlinearity, multimodality, uncertainty, and constraint, and beyond. *Front Inform Technol Electron Eng*, 18(12): 1913-1939. https://doi.org/10.1631/FITEE.1700379

Li TC, Wang XX, Liang Y, et al., 2020. On arithmetic average fusion and its application for distributed multi-Bernoulli multitarget tracking. *IEEE Trans Signal Process*, 68: 2883-2896. https://doi.org/10.1109/TSP.2020.2985643

Liang S, Zhu Y, Hao L, et al., 2019. Nearest-neighbour joint probabilistic data association filter based on random finite set. Proc 8th Int Conf on Control, Automation and Information Sciences, p.1-6. https://doi.org/10.1109/iccais46528.2019.9074585

Mahler RPS, 2003. Multitarget Bayes filtering via first-order multitarget moments. *IEEE Trans Aerosp Electron Syst*, 39(4):1152-1178. https://doi.org/10.1109/TAES.2003.1261119

Mahler RPS, 2007a. PHD filters of higher order in target number. *IEEE Trans Aerosp Electron Syst*, 43(4):1523-1543. https://doi.org/10.1109/TAES.2007.4441756

Mahler RPS, 2007b. Statistical Multisource-Multitarget Information Fusion. Artech House, Boston, USA.

Panakkal VP, Velmurugan R, 2013. Effective joint probabilistic data association using maximum a posteriori estimates of target states. Proc 16th Int Conf on Information Fusion, p.781-788.

Reid D, 1979. An algorithm for tracking multiple targets. *IEEE Trans Autom Contr*, 24(6):843-854. https://doi.org/10.1109/TAC.1979.1102177

Reuter S, Wilking B, Wiest J, et al., 2013. Real-time multi-object tracking using random finite sets. *IEEE Trans Aerosp Electron Syst*, 49(4):2666-2678. https://doi.org/10.1109/TAES.2013.6621844

Reuter S, Vo BT, Vo BN, et al., 2014. The labeled multi-Bernoulli filter. *IEEE Trans Signal Process*, 62(12):3246-3260. https://doi.org/10.1109/TSP.2014.2323064

Schuhmacher D, Vo BT, Vo BN, 2008. A consistent metric for performance evaluation of multi-object filters. *IEEE Trans Signal Process*, 56(8):3447-3457. https://doi.org/10.1109/TSP.2008.920469

Sidenbladh H, 2003. Multi-target particle filtering for the probability hypothesis density. Proc 6th Int Conf on Information Fusion, p.800-806. https://doi.org/10.1109/ICIF.2003.177321

Sidenbladh H, Wirkander SL, 2003. Tracking random sets of vehicles in terrain. Proc Conf on Computer Vision and Pattern Recognition Workshop, p.98. https://doi.org/10.1109/CVPRW.2003.10097

Svensson D, Svensson L, Guerriero M, et al., 2011. The multi-target set JPDA filter with target identity. *SPIE*, 8050: 805010. https://doi.org/10.1117/12.886946

Svensson L, Svensson D, Guerriero M, et al., 2011. Set JPDA filter for multitarget tracking. *IEEE Trans Signal Process*, 59(10):4677-4691. https://doi.org/10.1109/TSP.2011.2161294

Touri B, Nedić A, 2011. Alternative characterization of ergodicity for doubly stochastic chains. Proc 50th IEEE Conf on Decision and Control and European Control Conf, p.5371-5376. https://doi.org/10.1109/CDC.2011.6161372

Vo BN, Ma WK, 2006. The Gaussian mixture probability hypothesis density filter. *IEEE Trans Signal Process*, 54(11):4091-4104. https://doi.org/10.1109/TSP.2006.881190

Vo BN, Vo BT, 2019. A multi-scan labeled random finite set model for multi-object state estimation. *IEEE Trans Signal Process*, 67(19):4948-4963. https://doi.org/10.1109/TSP.2019.2928953

Vo BN, Singh S, Doucet A, 2005. Sequential Monte Carlo methods for multitarget filtering with random finite sets. *IEEE Trans Aerosp Electron Syst*, 41(4):1224-1245. https://doi.org/10.1109/TAES.2005.1561884

Vo BN, Vo BT, Phung D, 2014. Labeled random finite sets and the Bayes multi-target tracking filter. *IEEE Trans Signal Process*, 62(24):6554-6567. https://doi.org/10.1109/TSP.2014.2364014

Vo BT, Vo BN, 2013. Labeled random finite sets and multi-object conjugate priors. *IEEE Trans Signal Process*, 61(13):3460-3475. https://doi.org/10.1109/TSP.2013.2259822

Vo BT, Vo BN, Cantoni A, 2009. The cardinality balanced multi-target multi-Bernoulli filter and its implementations. *IEEE Trans Signal Process*, 57(2):409-423. https://doi.org/10.1109/TSP.2008.2007924

Williams JL, 2015. An efficient, variational approximation of the best fitting multi-Bernoulli filter. *IEEE Trans Signal Process*, 63(1):258-273. https://doi.org/10.1109/TSP.2014.2370946

Zajic T, Mahler RPS, 2003. Particle-systems implementation of the PHD multitarget-tracking filter. *SPIE*, 5096:291-299. https://doi.org/10.1117/12.488533

Zhu Y, Wang J, Liang S, 2017. Efficient joint probabilistic data association filter based on Kullback–Leibler divergence for multi-target tracking. *IET Radar Sonar Navig*, 11(10):1540-1548. https://doi.org/10.1049/iet-rsn.2017.0102

Zhu Y, Wang J, Liang S, 2019. Covariance control joint integrated probabilistic data association filter for multi-target tracking. *IET Radar Sonar Navig*, 13(4):584-592. https://doi.org/10.1049/iet-rsn.2018.5142