

基于DP-TBD的分布式异步迭代滤波融合算法研究

李洋洋* 李雯 易伟 孔令讲

(电子科技大学电子工程学院 成都 611731)

摘要: 该文主要运用检测前跟踪动态规划(Dynamic Programming-Track Before Detect)算法解决目标跟踪问题。动态规划(Dynamic Programming, DP)是一种通过对量测空间栅格化处理, 然后对离散的量测空间中所有可能的物理路径进行遍历的算法。然而, 该算法提供的是一种未经滤波和平滑的点迹序列。随着实际战争环境日益复杂, 基于单雷达的DP-TBD算法在信噪比(SNR)较低时跟踪效果不佳。此外, 由于DP-TBD算法没有状态误差协方差矩阵, 因此无法将不同雷达的点迹序列进行融合。而且由于通信时延和不同的采样周期, 不同雷达的数据往往是异步的。为了解决以上问题, 该文提出了一种基于DP-TBD的分布式异步迭代滤波融合算法(Dynamic Programming Fusion, DPF)。该算法分为两步, 第1步提出了一种迭代滤波方法对DP点迹进行处理; 第2步将不同雷达获得的异步状态估计转化为同步的, 接着利用几种分布式的融合方法来获取融合之后的状态估计。仿真结果说明, 和单雷达相比, 该融合算法可以有效提升目标跟踪的性能, 同时, 该算法也可以降低航迹丢失率和计算量。

关键词: 检测前跟踪; 动态规划; 迭代滤波; 多传感器; 分布式异步融合

中图分类号: TP212

文献标识码: A

文章编号: 2095-283X(2018)02-0254-09

DOI: 10.12000/JR17057

引用格式: 李洋洋, 李雯, 易伟, 等. 基于DP-TBD的分布式异步迭代滤波融合算法研究[J]. 雷达学报, 2018, 7(2): 254-262. DOI: 10.12000/JR17057.

Reference format: Li Yangyang, Li Wen, Yi Wei, *et al.*. A distributed asynchronous recursive filtering fusion algorithm via DP-TBD[J]. *Journal of Radars*, 2018, 7(2): 254-262. DOI: 10.12000/JR17057.

A Distributed Asynchronous Recursive Filtering Fusion Algorithm via DP-TBD

Li Yangyang Li Wen Yi Wei Kong Lingjiang

(School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China)

Abstract: In this paper, we address target tracking problems by the use of multiple sensors via the Dynamic Programming (DP)-based Track-Before-Detect (TBD) method. Generally, DP-TBD is a grid-based method that estimates target trajectories by searching all the physically admissible paths in a determinate discrete state space. However, this multi-frame detection algorithm provides plot sequences without filtering or smoothing. With the growing complexity of the battle field environment, single radar based on DP-TBD cannot achieve satisfactory results when the Signal-to-Noise Ratio (SNR) is low. Besides, it is very difficult to fuse plot sequences from different radars because they contain no state error covariance matrix. Furthermore, various radars always contain asynchronous data due to the diversity of sampling times and communication delays. To alleviate these problems, we propose a distributed asynchronous recursive filtering fusion (Dynamic Programming Fusion, DPF) algorithm based on DP-TBD, which is divided into two steps. In the first step, we propose an iterative filter algorithm via DP-TBD. Then, we convert the asynchronous evaluation data into

收稿日期: 2017-06-14; 改回日期: 2017-07-31; 网络出版: 2017-08-22

*通信作者: 李洋洋 575630861@qq.com

基金项目: 长江学者奖励计划, 中央高校基本科研基金(ZYGX2016J031), 中国博士后科学基金面上基金(2014M550465)和特别资助基金(2016T90845)

Foundation Items: The Chang Jiang Scholars Program, The Fundamental Research Funds of Central Universities under Grants (ZYGX2016J031), The Chinese Postdoctoral Science Foundation under Grant (2014M550465) and Special Grant (2016T90845)

synchronous data and implement several distributed fusion algorithms to estimate the target state. Simulation results show that the proposed algorithm can correctly estimate target trajectories and significantly enhance tracking accuracy compared to solo radar. In addition, this algorithm can decrease the track loss rate and calculation burden.

Key words: Track-Before-Detect (TBD); Dynamic Programming (DP); Recursive filtering; Multiple sensors; Distributed asynchronous fusion

1 引言

与传统跟踪方法相比^[1-3], 基于文献[4]中的检测前跟踪算法可以通过直接处理多帧原始数据对低信噪比下的目标进行更精确的跟踪。近年来, 文献[5-9]对基于动态规划的检测前跟踪算法进行了深入的研究。此外, 针对多目标的动态规划检测前跟踪技术(Dynamic Programming-Track Before Detect, DP-TBD)也在很多文献中也得到了广泛关注^[10,11], 检测前跟踪算法在实际雷达系统中的各种应用^[12-14]也得到了普遍关注。另外由于巨大的计算量限制了DP-TBD算法的应用, 一些新颖地降低计算量的方法^[15,16]也得到了深入探讨。

与传统跟踪直接提供离散的点迹不同, DP-TBD算法获取的是一些没有经过滤波和平滑的短航迹。在本文当中, 作者称它为点迹序列。因此, 传统跟踪的航迹关联和滤波算法不再适用于DP-TBD。同时, DP-TBD算法是一种栅格划分的方法, 导致了该算法至少有半个栅格的损失。低信噪比(Signal-to-Noise Ratio, SNR)和不正确的状态转移区域都会导致航迹丢失, 因此迫切地需要一种针对DP-TBD的滤波算法来提升点迹的跟踪精度。

文献[17-19]提出了基于DP-TBD的各种迭代滤波算法。然而, 随着战场环境的日益严峻, 单个传感器无法满足目标跟踪的精度需求。因此, 多个传感器协同探测势在必行。为了利用多个传感器的信息, 需要将各个传感器的观测信息进行正确的融合。当今的数据融合方法主要分为集中式和分布式融合。近年来, 集中式融合取得了广泛的关注^[20,21], 因为集中式融合是将量测数据直接送到融合中心, 没有任何数据损失, 所以它是一种最优的融合策略。然而在这种情况下, 融合中心会承担巨大的计算量, 尤其是传感器数量增加时, 对于融合中心来说计算量往往是灾难性的。因此导致了集中式融合的系统鲁棒性不高。至于分布式融合^[23-25], 是将单个传感器的一系列估计信息送到融合中心。通过这种方法, 不仅可以提升系统的鲁棒性, 还可以大大降低融合中心的计算负担。但是基于DP-TBD的分布式融合文献较少^[26]。此外, 该文献也并没有考虑数据异步问题和滤波问题。

虽然分布式融合取得了丰硕成果, 但是大部分工作都是基于相同采样率的前提下, 然而这种情况在实际中十分少见。这是因为实际中传感器往往是有不同的采样周期和初始采样时刻^[27-29]。因此怎样转换异步估计数据也成为主要问题之一。

为了解决以上所有问题, 本文提出了一种基于DP-TBD的分布式异步迭代滤波融合(Dynamic Programming Fusion, DPF)算法。首先将单个传感器的估计状态进行迭代滤波; 然后, 我们转换异步的估计数据为同步的估计数据; 最后, 运用几种分布式的融合准则来获取融合之后的目标状态估计。通过上述方法可以显著提升对微弱目标的检测跟踪性能。

2 量测与目标模型建立

2.1 目标模型

假定在检测区域内有 P 个传感器, 每个传感器采样 F' 帧。考虑一个点目标在2维平面内运动, 因此目标运动状态方程是建立在4维的状态空间, 定义为:

$$\mathbf{x}_k = [x_k \quad \dot{x}_k \quad y_k \quad \dot{y}_k]^T, \quad 1 \leq k \leq P \times F' \quad (1)$$

其中, $[\cdot]^T$ 代表矩阵的转置, x_k, y_k 表示目标在第 k 帧的位置元素; \dot{x}_k, \dot{y}_k 表示第 k 帧目标的速度元素。目标的状态方程式为:

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{v}_{k-1}, \quad 1 \leq k \leq P \times F' \quad (2)$$

其中, 状态转移矩阵 \mathbf{F} 定义为:

$$\mathbf{F} = \begin{bmatrix} \mathbf{E}_s & 0 \\ 0 & \mathbf{E}_s \end{bmatrix}, \quad \mathbf{E}_s = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \quad (3)$$

式中, T 代表两帧之间的时间间隔。过程噪声 \mathbf{v}_k 的协方差矩阵定义为:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{q} & 0 \\ 0 & \mathbf{q} \end{bmatrix}, \quad \mathbf{q} = q_s \begin{bmatrix} T^3/3 & T^2/2 \\ T^2/2 & T \end{bmatrix} \quad (4)$$

其中, q_s 是检测空间的加速度噪声功率谱密度。

2.2 量测模型

对第 p 个雷达在 k 时刻的量测模型定义为:

$$\mathbf{z}_{pk} = \mathbf{H}\mathbf{x}_{pk} + \mathbf{w}_{pk}, \quad k = 1, 2, \dots, P \times F' \quad (5)$$

其中

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (6)$$

量测噪声 $w_{pk} = [w_{pk,x}, w_{pk,y}]$ 与过程噪声独立，它的协方差是 R_{pk} 。 $w_{pk,x}$ 和 $w_{pk,y}$ 均为零均值方差为1的高斯分布噪声且相互独立，协方差矩阵 $R_{pk} = E[w_{pk}w_{pk}']$ 。观测噪声协方差的大小由第 k 帧的 DP 检测概率 P_{pD}^k 决定， $p(w_{pk,x})|_{w_{pk,x}=0} = p(w_{pk,y})|_{w_{pk,y}=0} = \sqrt{P_{pD}^k}$ 。

为了取得实时量测，本算法采用滑窗处理的方法，滑窗长度选为 N 。如图1，我们对一次滑窗过程中的时间 k 从1到 N 编号，如第 $k-N+1$ 帧编号为第1帧，那么第 k 帧编号为 N 。因此对于雷达 p ， k 时刻的一次滑窗量测定义为：

$$Z_{p(k-N+1:k)}^k = \{z_{pi}^k, i = k - N + 1, \dots, k\} \quad (7)$$

其中， z_{pi}^k 中上标 k 代表 k 时刻，下标 i 从 $k-N+1$ 到 k 代表 k 时刻的一次滑窗量测。并且 $Z_{p(k-N+1:k)}^k$ 代表该次滑窗中的多帧量测集合。

3 算法提出

算法流程分为：航迹起始、DP迭代、航迹关联、迭代滤波、异步状态转化和DPF融合。算法步骤的详细流程图和每一步之间的关系如图2所示。

$$P_{pk}^{N-1} = \begin{bmatrix} R_{pk}(1,1) & R_{pk}(1,1)/T_p & 0 & 0 \\ R_{pk}(1,1)/T & 2R_{pk}(1,1)/T_p^2 & 0 & 0 \\ 0 & 0 & R_{pk}(2,2) & R_{pk}(2,2)/T_p \\ 0 & 0 & R_{pk}(2,2)/T_p & 2R_{pk}(2,2)/T_p^2 \end{bmatrix} \quad (9)$$

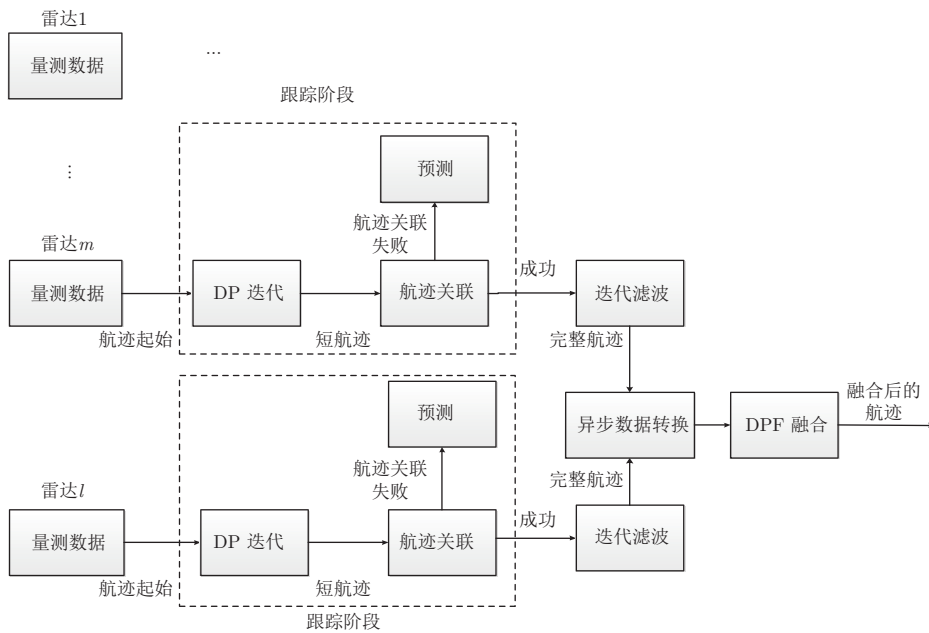


图 2 DPF算法详细流程图

Fig. 2 The detail method flow of DPF algorithm

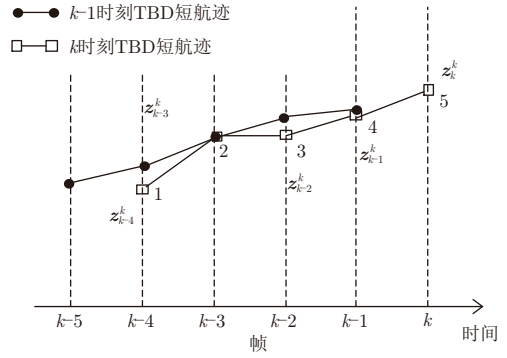


图 1 当 $N=5$ 时滑窗间多帧关系示意图

Fig. 1 The relationships of multiple sliding window frames when $N=5$

3.1 航迹起始

对于一次滑窗当中的第 n 帧，传感器 p 的检测概率为 P_{pD}^n [9] ($n = 1, 2, \dots, N, p = 1, 2, \dots, P$)，采样周期是 T_p 。对于一次DP中的所有帧 n ($n = 1, 2, \dots, N$)， R_{pn} 表示为：

$$R_{pn} = \begin{bmatrix} 1/(2\pi P_{pD}^n) & 0 \\ 0 & 1/(2\pi P_{pD}^n) \end{bmatrix} \quad (8)$$

对于传感器 p ，目标前 $N-1$ 帧的状态估计 x_{pk}^{N-1} ($k = 1, 2, \dots, N-1$) 由目标真实运动状态量化。此外，状态估计误差协方差矩阵初始化为：

其中, $\mathbf{R}_{pk}(1,1)$ 代表 \mathbf{R}_{pk} 矩阵里面的第(1,1)中的元素, 并且初始化时间 $k=N$ 。

3.2 跟踪阶段

利用 DP-TBD 算法处理一批回波量测 $\mathbf{Z}_{p(k-N+1:k)}^k$ 并且得到目标一次 DP 的运动状态估计 $\hat{\mathbf{x}}_{p(k-N+1)}^k, \hat{\mathbf{x}}_{p(k-N+2)}^k, \dots, \hat{\mathbf{x}}_{pk}^k$, 这里的 $\hat{\mathbf{x}}_{pi}^k (i = k - N + 1, \dots, k)$ 代表对第 p 个传感器在滑窗 k 时刻第 i 帧的运动状态估计。DP-TBD 算法的详细步骤在文献[4-9]有详细阐释, 因为篇幅原因, 这里不再阐述。然后, 进行点迹序列的航迹关联。

$$\hat{\mathbf{x}}_{pi}^k - \hat{\mathbf{x}}_{p(i+1)}^{k-1} < G, \quad i = k - N + 1, \dots, k - 1 \quad (10)$$

其中, G 是航迹关联的门限。当 DP 迭代的结果关联成功的时候, 用迭代滤波器来处理这些点迹序列。

3.3 迭代滤波器

对于第 p 个传感器, 令它的椭圆波门门限为 γ_p 。迭代滤波器处理过程可以分为两步, 第1步, 对传感器 p 从 $i = (k - N + 1)$ 到 $(k - 1)$ 帧的量测 $\mathbf{Z}_{p(k-N+1:k-1)}^k$ 进行滤波处理。首先计算它的新息过程:

$$\alpha_{pi}^k = \mathbf{z}_{pi}^k - \mathbf{H} \hat{\mathbf{x}}_{pi}^{k-1} \quad (11)$$

接着, 定义一步预测误差的自相关矩阵为:

$$\mathbf{S}_{pi}^k = \mathbf{H} \mathbf{P}_{pi}^{k-1} \mathbf{H}' + \mathbf{R}_{p(N-k+i)} \quad (12)$$

之后, 计算迭代滤波增益:

$$\mathbf{B}_{pi}^k = \mathbf{P}_{pi}^{k-1} \mathbf{H}' \left(\mathbf{S}_{pi}^k \right)^{-1} \quad (13)$$

最后更新状态误差协方差矩阵:

$$\mathbf{P}_{pi}^k = \mathbf{P}_{pi}^{k-1} - \mathbf{B}_{pi}^k \mathbf{H} \mathbf{P}_{pi}^{k-1} \quad (14)$$

如果

$$\left(\alpha_{pi}^k \right)' \left(\mathbf{S}_{pi}^k \right)^{-1} \alpha_{pi}^k \leq \gamma_p \quad (15)$$

则对状态进行更新

$$\hat{\mathbf{x}}_{pi}^k = \hat{\mathbf{x}}_{pi}^{k-1} + \mathbf{B}_{pi}^k \alpha_{pi}^k \quad (16)$$

否则

$$\hat{\mathbf{x}}_{pi}^k = \hat{\mathbf{x}}_{pi}^{k-1} \quad (17)$$

如果在这次 DP 迭代当中没有检测结果, 即没有值函数超过门限或者航迹关联失败。那么就只需要计算式(12), 式(13), 式(14)和式(17)。

第2步, 通过迭代滤波器去处理第 k 帧的量测 \mathbf{z}_{pk}^k 。

$$\alpha_{pk}^k = \mathbf{z}_{pk}^k - \mathbf{H} \mathbf{F} \hat{\mathbf{x}}_{p(k-1)}^k \quad (18)$$

$$\mathbf{S}_{pk}^k = \mathbf{H} \left(\mathbf{F} \mathbf{P}_{p(k-1)}^k \mathbf{F}' + \mathbf{Q} \right) \mathbf{H}' + \mathbf{R}_{pN} \quad (19)$$

$$\mathbf{B}_{pk}^k = \left(\mathbf{F} \mathbf{P}_{p(k-1)}^k \mathbf{F}' + \mathbf{Q} \right) \mathbf{H}' \left(\mathbf{S}_{pk}^k \right)^{-1} \quad (20)$$

$$\mathbf{P}_{pk}^k = \left(\mathbf{F} \mathbf{P}_{p(k-1)}^k \mathbf{F}' + \mathbf{Q} \right) - \mathbf{B}_{pk}^k \mathbf{H} \left(\mathbf{F} \mathbf{P}_{p(k-1)}^k \mathbf{F}' + \mathbf{Q} \right) \quad (21)$$

如果

$$\left(\alpha_{pk}^k \right)' \left(\mathbf{S}_{pk}^k \right)^{-1} \alpha_{pk}^k \leq \gamma_p \quad (22)$$

则对状态进行更新

$$\hat{\mathbf{x}}_{pk}^k = \mathbf{F} \hat{\mathbf{x}}_{p(k-1)}^k + \mathbf{B}_{pk}^k \alpha_{pk}^k \quad (23)$$

否则进行一步预测

$$\hat{\mathbf{x}}_{pk}^k = \mathbf{F} \hat{\mathbf{x}}_{p(k-1)}^k \quad (24)$$

如果在这次 DP 积累中没有检测结果或者航迹关联失败, 那么只需要计算式(19), 式(20), 式(21)和式(24)。然后, 利用滑窗算法来处理这些点迹序列直到得到一条完整的航迹。

3.4 异步数据转换

正如上文所述, 通过这种方法, 我们可以得到一条完整航迹和它的状态误差协方差矩阵。为了描述的方便, 我们定义拥有最长采样周期的传感器为 l , 它在第 q 帧的估计状态和状态误差协方差矩阵分别为 $\hat{\mathbf{x}}_{lq}$ 和 $\mathbf{P}_{lq} (1 \leq q \leq F')$ 。其余的任意一个传感器 m 的状态估计为 $\hat{\mathbf{x}}_{mz}$, 状态误差协方差矩阵为 $\mathbf{P}_{mz} (m = 1, 2, \dots, l - 1, l + 1, \dots, P, 1 \leq z \leq F')$ 。

首先, 我们计算状态 \mathbf{x}_{lq} 和 \mathbf{x}_{mz} 的时间差

$$\Delta_{qz}^{lm} = T_{lq} - T_{mz} \quad (25)$$

其中, T_{lq} 和 T_{mz} 分别代表传感器 l 的第 q 帧和传感器 m 的第 z 帧。接着, 我们利用 Δ_{qz}^{lm} 来进行状态预测和状态误差协方差矩阵的预测:

$$\hat{\mathbf{x}}_{lq}^m = \mathbf{F}_{qz} \times \mathbf{x}_{mz} \quad (26)$$

$$\mathbf{P}_{lq}^m = \mathbf{Q}_{qz} + \mathbf{F}_{qz} \times \mathbf{P}_{mz} \times (\mathbf{F}_{qz})^T \quad (27)$$

其中, \mathbf{F}_{qz} 和 \mathbf{Q}_{qz} 分别由式(3)和式(4)中 Δ_{qz}^{lm} 决定的。转化之后, $\hat{\mathbf{x}}_{lq}^m$ 和 \mathbf{P}_{lq}^m 是传感器 m 对传感器 l 采样时刻的状态估计和状态误差协方差矩阵预测。转换详细步骤如图3所示。如果

$$\Delta_{qz}^{lm} = 0 \quad (28)$$

那么我们就跳过式(26)和式(27)。

为了利用传感器 m 的所有量测信息来获取更精确的目标状态估计。如果不同的状态 $\hat{\mathbf{x}}_{m1}, \hat{\mathbf{x}}_{m2}, \dots, \hat{\mathbf{x}}_{mw}$ 转换到了同一时刻 T_{lq} , 我们就利用式(29)和式(30)来计算 $\hat{\mathbf{x}}_{lq}^m$ 和 \mathbf{P}_{lq}^m ,

$$\hat{\mathbf{x}}_{lq}^m = (\hat{\mathbf{x}}_{m1} + \hat{\mathbf{x}}_{m2} + \dots + \hat{\mathbf{x}}_{mw}) / w \quad (29)$$

$$\mathbf{P}_{lq}^m = (\mathbf{P}_{m1} + \mathbf{P}_{m2} + \dots + \mathbf{P}_{mw}) / w \quad (30)$$

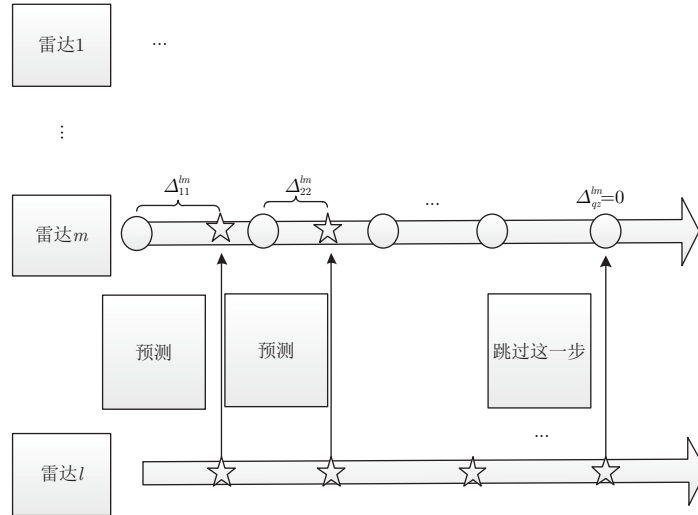


图 3 异步数据转换示意图

Fig. 3 Asynchronous data conversion figure

最后, 我们利用上文所述的方法来处理每一个传感器的状态估计和误差协方差估计并且均对齐到传感器 l 的采样时刻下, 之后选择融合时刻为传感器的采样时刻。

3.5 DPF融合

通过上文处理之后, 目标状态估计 \hat{x}_{lq}^l 和 \hat{x}_{lq}^m 就可以基于一系列的融合准则进行融合。在本文中, 我们利用基于状态误差协方差矩阵最小迹的融合方式(TCI-DPF, DPF algorithm based on minimizing the Trace of Covariance Intersection criterion)和简单凸组合的融合方式(SS-DPF, DPF algorithm based on Simple convex combination criterion)进行融合^[30,31]。

TCI-DPF表示为:

$$\hat{x}_q^{\text{TCI}} = \mathbf{P}_q^{\text{TCI}} \left(w_{lq}^1 (\mathbf{P}_{lq}^1)^{-1} \hat{x}_{lq}^1 + \dots + w_{lq}^l (\mathbf{P}_{lq}^l)^{-1} \hat{x}_{lq}^l + \dots + w_{lq}^P (\mathbf{P}_{lq}^P)^{-1} \hat{x}_{lq}^P \right) \quad (31)$$

其中

$$\mathbf{P}_q^{\text{TCI}} = (w_{lq}^1 (\mathbf{P}_{lq}^1)^{-1} + \dots + w_{lq}^l (\mathbf{P}_{lq}^l)^{-1} + \dots + w_{lq}^P (\mathbf{P}_{lq}^P)^{-1})^{-1} \quad (32)$$

$$w_{lq}^1 + \dots + w_{lq}^l + \dots + w_{lq}^P = 1 \quad (33)$$

并且 $w_{lq}^1, \dots, w_{lq}^P$ 取决于 $\min \text{trace}(\mathbf{P}_q^{\text{TCI}})$, trace 代表行列式的迹。至于SS-DPF融合方法, 它满足式(31)和式(32), 但是 $w_{lq}^1, \dots, w_{lq}^P$ 均等于 $1/P$ 。

3.6 小结

通过上文所述的步骤, 我们可以实现DPF算法。这种算法通过利用多个传感器的信息进行协同探测, 可以有效地提升弱目标检测和跟踪的性能。这种方法也可以适用于一些其他的批处理TBD算法, 比如霍夫变换等。此外, 虽然本文是基于单目

标的推导方法, 但是我们可以利用文献[10,11]的方法来获取多目标跟踪的方法。

4 仿真实验

在本小节中, 我们运用仿真来证明DPF算法的有效性。考虑一个恒定幅度的点目标在一个2维的笛卡尔坐标系下运动, 观测区域由 200×200 个单元格构成。运动规律遵从式(2)且噪声功率谱密度 $q_s = 0.001$ 。假设有两个传感器在跟踪目标且DP-TBD算法的门限是基于恒虚警0.01, 传感器1和传感器2的采样周期分别是1 s和2 s。此外, 一次DP迭代处理6帧, 总处理帧数为20帧。门限区域是99.97%且 $\gamma_p = 16$ 。航迹起始是由离散化目标前 $N-1$ 帧实际状态得到的。

图4表示运用SS-DPF算法和TCI-DPF算法单次蒙特卡洛仿真下的目标航迹。和单个传感器相比, 由于有融合的处理, 我们可以在某些帧看到, SS-DPF和TCI-DPF拥有更小的估计误差且可以得到更精确的估计效果。

我们利用参数RMSE位置误差和航迹丢失率来评价这个算法的可行性。航迹丢失率^[32]是在整个跟踪过程中, 如果估计的误差超过了一个低门限 c_1 , 那么这个估计器继续等 N 帧的数据来判断它是否降到了 c_1 以下, 如果降到了那我们认为这仍然是一条有效的航迹, 如果没有, 那么我们就认为航迹丢失了。此外, 如果这个误差超过了一个高门限 c_2 , 我们就直接认为航迹丢失了。在本次仿真中, c_1 和 c_2 分别设为2和4并且蒙特卡洛仿真次数设置为200次。

图5表示两种算法在SNR=8 dB和SNR=12 dB时的RMSE。由图可知, DPF算法的跟踪性能远好

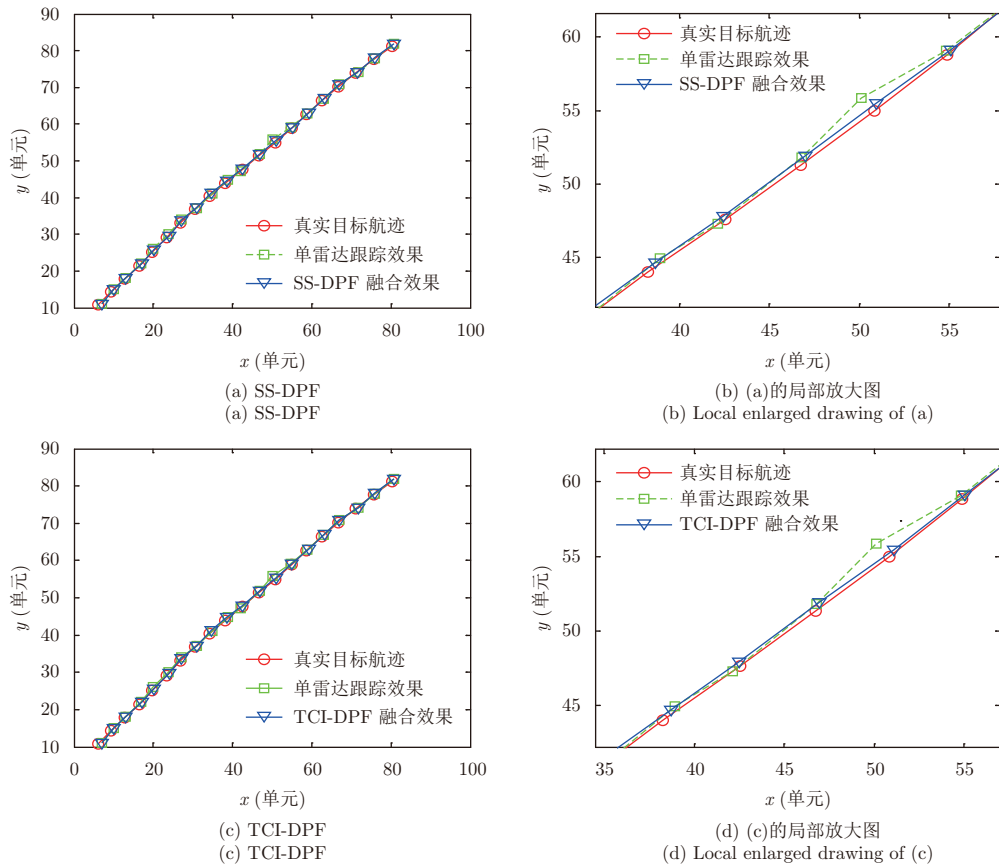


图4 当SNR=8时的一次蒙特卡洛仿真结果

Fig. 4 One time Monte Carlo simulation result when SNR=8

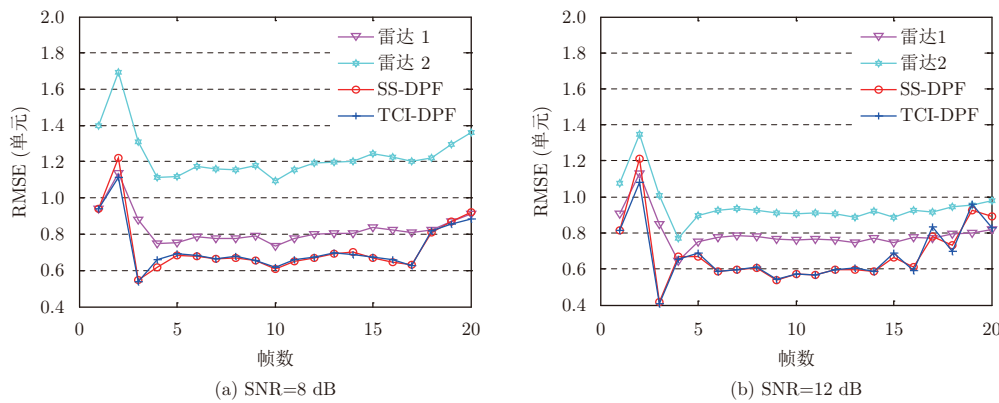


图5 从第1帧到第20帧的RMSE

Fig. 5 RMSE from the 1st frame to the 20nd frame

于单个传感器。当信噪比为8 dB时，传感器1和传感器2的RMSE分别是0.8293和1.2350。经过TCI融合后，每帧的RMSE是0.7238，与传感器1比提升了12.72%。SS-DPF融合方法平均每帧的RMSE是0.7269，和传感器1相比降低了12.35%。当信噪比为12 dB时，传感器1和传感器2的平均RMSE是0.7954和0.9475，TCI-DPF方法可以降低14.99%的RMSE到0.6761。至于SS-DPF方法，RMSE是0.6816，和传感器1相比减小了14.30%。

图6展示了DPF算法和传统跟踪分布式融合算法(TDF)的航迹丢失率，在这里 c_1 和 c_2 分别取为3和8，其余仿真条件和上文所述相同。TDF算法是一种检测后跟踪的方法^[31]，它的步骤包括恒虚警检测(Constant False Alarm Rate, CFAR)，航迹起始，航迹关联，卡尔曼滤波和基于融合准则的分布式融合^[30]，两者融合均基于TCI准则。结果显示在同样的信噪比条件下，和TDF方法相比，DPF方法可大大减少航迹丢失率。如当信噪比为10 dB

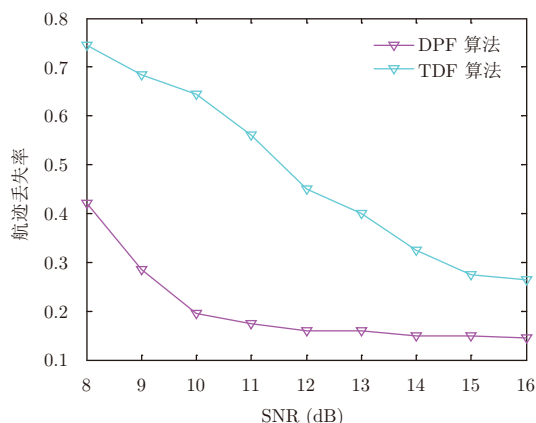


图 6 信噪比从8 dB到16 dB, DPF算法和TDF算法的航迹丢失率
Fig. 6 Track loss rate of DPF algorithm and TDF method from SNR=8 dB to SNR=16 dB

的时候, DPF方法的航迹丢失率约为20%, 相比TDF方法的65%降低了超过200%的丢失率。这是因为在低信噪比环境下, TDF算法的CFAR会滤除到很多目标的信息, 导致大量的航迹丢失。但是DPF算法就是一种解决该问题的有效方法。当信噪比提升时, 两个算法的航迹丢失率的差距减小。这个仿真证明了DPF方法在减小航迹丢失率方面的优秀表现。

最后, DPF算法和CDPF算法的一次DP迭代的执行时间如表1所示。CDPF算法是一种基于DP-TBD的集中式异步融合方法^[33]。这种方法是直接传送所有的量测数据到融合中心, 然后融合中心进行DP处理的一种方法。结果显示DPF算法和CDPF方法的总执行时间差不多, 但DPF方法的计算负担分散在各个传感器上。至于CDPF算法, 由于单个传感器只需要将它的量测数据传送至融合中心, 因此导致该算法的所有计算负担都承载在融合中心, 尤其是当传感器数目增加时, 融合中心的计算量巨大, 这会导致系统的鲁棒性降低。而且融合中心一旦崩溃, 就会导致整个系统崩溃。但DPF算法把计算量近似于均匀地分配给了每个雷达, 这样即使单个传感器出现问题, 也不会导致整个系统崩溃。

表 1 DPF算法和CDPF算法一次DP迭代的执行时间(s)

Tab. 1 One time execution time of DPF algorithm and CDPF algorithm (s)

参数	DPF算法	CDPF算法
传感器1的跟踪时间	4.07	0.052
传感器2的跟踪时间	3.661	0.043
融合时间	0.1	7.74

5 结束语

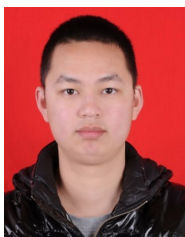
在本文中, 通过利用不同传感器的估计信息, 提出了基于DP-TBD的分布式异步迭代滤波融合算法, 并且解决了多传感器融合问题。仿真结论说明了基于检测前跟踪动态规划分布式异步融合算法的有效性。并且相对于单个传感器来说, 该算法大大改善了对弱目标跟踪的性能。此外也降低了航迹丢失率并且提高了系统的鲁棒性。

参考文献

- [1] Bar-Shalom Y and Li Xiao-rong. Multitarget-Multisensor Tracking: Principles and Techniques[M]. Storrs, CT: YBS, 1995.
- [2] Bar-Shalom Y and Blair W D. Multitarget-Multisensor Tracking: Applications and Advances, Vol. III[M]. Norwood, MA: Artech House, 2000.
- [3] Bar-Shalom Y, Daum F, and Huang J. The probabilistic data association filter[J]. *IEEE Control Systems*, 2009, 29(6): 82–100. DOI: [10.1109/MCS.2009.934469](https://doi.org/10.1109/MCS.2009.934469).
- [4] Davey S J and Rutten M G. A comparison of three algorithms for tracking dim targets[C]. Proceedings of 2007 IEEE Information, Decision and Control, Adelaide, Australia, 2007: 342–347.
- [5] Barniv Y. Dynamic programming solution for detecting dim moving targets[J]. *IEEE Transactions on Aerospace and Electronic Systems*, 1985, AES-21(1): 144–156. DOI: [10.1109/TAES.1985.310548](https://doi.org/10.1109/TAES.1985.310548).
- [6] Barniv Y and Kella O. Dynamic programming solution for detecting dim moving targets part II: Analysis[J]. *IEEE Transactions on Aerospace and Electronic Systems*, 1987, AES-23(6): 776–788. DOI: [10.1109/TAES.1987.310914](https://doi.org/10.1109/TAES.1987.310914).
- [7] Arnold J, Shaw S W, and Pasternack H. Efficient target tracking using dynamic programming[J]. *IEEE Transactions on Aerospace and Electronic Systems*, 1993, 29(1): 44–56. DOI: [10.1109/7.249112](https://doi.org/10.1109/7.249112).
- [8] Tonissen S M and Evans R J. Performance of dynamic programming techniques for Track-Before-Detect[J]. *IEEE Transactions on Aerospace and Electronic Systems*, 1996, 32(4): 1440–1451. DOI: [10.1109/7.543865](https://doi.org/10.1109/7.543865).
- [9] Johnston L A and Krishnamurthy V. Performance analysis of a dynamic programming track before detect algorithm[J]. *IEEE Transactions on Aerospace and Electronic Systems*, 2002, 38(1): 228–242. DOI: [10.1109/7.993242](https://doi.org/10.1109/7.993242).
- [10] Yi Wei, Morelande M R, Kong Ling-jiang, et al.. An efficient multi-frame track-before-detect algorithm for multi-target tracking[J]. *IEEE Journal of Selected Topics in Signal Processing*, 2013, 7(3): 421–434. DOI: [10.1109/JSTSP.2013.2256415](https://doi.org/10.1109/JSTSP.2013.2256415).

- [11] Buzzi S, Lops M, Venturino L, *et al.* Track-before-detect procedures in a multi-target environment[J]. *IEEE Transactions on Aerospace and Electronic Systems*, 2008, 44(3): 1135–1150. DOI: [10.1109/TAES.2008.4655369](https://doi.org/10.1109/TAES.2008.4655369).
- [12] Wallace W R. The use of track-before-detect in pulse-Doppler radar[C]. Proceedings of RADAR 2002, Edinburgh, UK, 2002: 315–319.
- [13] Buzzi S, Lops M, and Venturino L. Track-before-detect procedures for early detection of moving target from airborne radars[J]. *IEEE Transactions on Aerospace and Electronic Systems*, 2005, 41(3): 937–954. DOI: [10.1109/TAES.2005.1541440](https://doi.org/10.1109/TAES.2005.1541440).
- [14] Orlando D, Ricci G, and Bar-Shalom Y. Track-before-detect algorithms for targets with Kinematic constraints[J]. *IEEE Transactions on Aerospace and Electronic Systems*, 2011, 47(3): 1837–1849. DOI: [10.1109/TAES.2011.5937268](https://doi.org/10.1109/TAES.2011.5937268).
- [15] Grossi E, Lops M, and Venturino L. A Novel dynamic programming algorithm for Track-Before-Detect in radar systems[J]. *IEEE Transactions on Signal Processing*, 2013, 61(10): 2608–2619. DOI: [10.1109/TSP.2013.2251338](https://doi.org/10.1109/TSP.2013.2251338).
- [16] Yi Wei, Kong Ling-jiang, and Yang Jian-yu. Thresholding process based dynamic programming Track-before-detect algorithm[J]. *IEICE Transactions on Communications*, 2013, E96.B(1): 291–300. DOI: [10.1587/transcom.E96.B.291](https://doi.org/10.1587/transcom.E96.B.291).
- [17] Liu Rui, Yi Wei, Kong Ling-jiang, *et al.* Recursive filtering for target tracking in multi-frame Track-Before-Detect[C]. Proceedings of the 2014 17th International Conference on Information Fusion (FUSION), Salamanca, Spain, 2014: 1–6.
- [18] Fang Zi-cheng, Yi Wei, and Kong Ling-jiang. A tracking approach for low observable target using plot-sequences of multi-frame detection[C]. Proceedings of the 2016 19th International Conference on Information Fusion, Heidelberg, Germany, 2016: 1427–1433.
- [19] Govaers F, Rong Yang, Chee L H, *et al.* Track-before-detect in distributed sensor applications[J]. *EURASIP Journal on Advances in Signal Processing*, 2011, 2011: 20. DOI: [10.1186/1687-6180-2011-20](https://doi.org/10.1186/1687-6180-2011-20).
- [20] Gao Xin-bo, Chen Jin-guang, Tao Da-cheng, *et al.* Multi-sensor centralized fusion without measurement noise covariance by variational Bayesian approximation[J]. *IEEE Transactions on Aerospace and Electronic Systems*, 2011, 47(1): 718–727. DOI: [10.1109/TAES.2011.5705702](https://doi.org/10.1109/TAES.2011.5705702).
- [21] Ma Jing and Sun Shu-li. Centralized fusion estimators for multisensor systems with random sensor delays, multiple packet dropouts and uncertain observations[J]. *IEEE Sensors Journal*, 2013, 13(4): 1228–1235. DOI: [10.1109/JSEN.2012.2227995](https://doi.org/10.1109/JSEN.2012.2227995).
- [22] Zhai Yan, Yeary M B, Havlicek J P, *et al.* A new centralized sensor fusion-tracking methodology based on particle filtering for power-aware systems[J]. *IEEE Transactions on Instrumentation and Measurement*, 2008, 57(10): 2377–2387. DOI: [10.1109/TIM.2008.919009](https://doi.org/10.1109/TIM.2008.919009).
- [23] Mohammadi A and Asif A. Distributed particle filter implementation with intermittent/irregular consensus convergence[J]. *IEEE Transactions on Signal Processing*, 2013, 61(10): 2572–2587. DOI: [10.1109/TSP.2013.2245123](https://doi.org/10.1109/TSP.2013.2245123).
- [24] Zhao Tong and Nehorai A. Distributed sequential Bayesian estimation of a diffusive source in wireless sensor networks[J]. *IEEE Transactions on Signal Processing*, 2007, 55(4): 1511–1524. DOI: [10.1109/TSP.2006.889975](https://doi.org/10.1109/TSP.2006.889975).
- [25] Üney M, Clark D E, and Julier S J. Distributed fusion of PHD filters via exponential mixture densities[J]. *IEEE Journal of Selected Topics in Signal Processing*, 2013, 7(3): 521–531. DOI: [10.1109/JSTSP.2013.2257162](https://doi.org/10.1109/JSTSP.2013.2257162).
- [26] Guo Yun-fei, Zeng Ze-bing, and Zhao Shang-yu. An amplitude association dynamic programming TBD algorithm with multistatic radar[C]. Proceedings of the 2016 35th Chinese Control Conference, Chengdu, China, 2016: 5076–5079.
- [27] Talebi H and Hemmatyar A. Asynchronous track-to-track fusion by direct estimation of time of sample in sensor networks[J]. *IEEE Sensors Journal*, 2014, 14(1): 210–217. DOI: [10.1109/JSEN.2013.2281394](https://doi.org/10.1109/JSEN.2013.2281394).
- [28] Hu Yanyan, Duan Zhansheng, and Zhou Donghua. Estimation fusion with general asynchronous multi-rate sensors[J]. *IEEE Transactions on Aerospace and Electronic Systems*, 2010, 46(4): 2090–2102. DOI: [10.1109/TAES.2010.5595618](https://doi.org/10.1109/TAES.2010.5595618).
- [29] Yan L P, Liu B S, and Zhou D H. Asynchronous multirate multisensor information fusion algorithm[J]. *IEEE Transactions on Aerospace and Electronic Systems*, 2007, 43(3): 1135–1146. DOI: [10.1109/TAES.2007.4383603](https://doi.org/10.1109/TAES.2007.4383603).
- [30] Wang Yi-min and Li X R. Distributed estimation fusion with unavailable cross-correlation[J]. *IEEE Transactions on Aerospace and Electronic Systems*, 2012, 48(1): 259–278. DOI: [10.1109/TAES.2012.6129634](https://doi.org/10.1109/TAES.2012.6129634).
- [31] Julier S J. An empirical study into the use of chernoff information for robust, distributed fusion of Gaussian mixture models[C]. Proceedings of the 2006 9th IEEE International Conference on Information Fusion, Florence, Italy, 2006: 1–8.
- [32] Aprile A, Grossi E, Lops M, *et al.* An application of Track-Before-Detect to sea-clutter rejection: Experimental results based on real data[C]. Proceedings of the 2014 11th European Radar Conference, Rome, Italy, 2014: 505–508.
- [33] Li Yang-yang, Wang Jing-he, Yi Wei, *et al.* A centralized asynchronous fusion algorithm for sensors with different resolution via DP-TBD[C]. Proceedings of 2017 IEEE Radar Conference, Seattle, USA, 2017: 922–927.

作者简介



李洋漾(1993-), 男, 四川人, 电子科技大学硕士研究生, 研究方向为多传感器数据融合理论、弱小目标检测跟踪技术。

E-mail: 575630861@qq.com



易 伟(1983-), 男, 四川人, 电子科技大学副教授, 研究方向为统计信号处理、雷达信号与数据处理、多传感器数据融合理论、弱小目标检测跟踪技术等。

E-mail: kussoyi@gmail.com



李 雯(1993-), 女, 陕西人, 电子科技大学硕士研究生, 研究方向为雷达信号处理、雷达通信一体化波形设计。

E-mail: 1757152507@qq.com



孔令讲(1974-), 男, 河南人, 博士, 电子科技大学教授, 研究方向为宽带雷达系统技术、弱目标检测跟踪技术、雷达协同探测技术、相控阵激光雷达技术, 科研概况: 主要承担国家863、国防预研、自然科学基金等科研项目。