基于消息传递的九维协作定位与导航（嵌入粒子流）

Lukas Wielandner*†、Erik Leitinger*、Florian Meyer‡、Klaus Witrisal*†

* 格拉茨工业大学
† Christian Doppler位置感知电子系统实验室
‡ 加利福尼亚大学圣迭戈分校

arXiv:2301.01173v1 [eess.SP] 3 Jan 2023

摘要——**协作定位（CL）是一项重要技术，可用于位置感知通信网络、现代便利服务和公共安全等创新服务。我们研究由移动智能体组成的无线网络，这些智能体通过彼此进行成对测量并交换位置信息来实现自我定位。置信传播（BP）是协作定位领域先进的贝叶斯方法。在协作定位中，常采用基于粒子的置信传播实现，以应对非线性测量模型和状态动态。然而，已知基于粒子的置信传播算法在具有高维状态的大型、密集移动智能体网络中容易出现粒子退化。**

本文通过粒子流推导了用于协同定位（CL）的置信传播（BP）消息，并据此开发了一种分布式基于粒子的消息传递算法，可避免粒子退化。我们结合粒子流与置信传播的方法，只需极少数量的粒子，即可为智能体状态计算出高度准确的提议分布。在准确度和运行时间方面，该方法优于传统的基于粒子的置信传播算法。此外，我们将所提方法与一种集中式粒子流实现（即精确的 Daum-Huang 滤波器）以及西格玛点置信传播进行比较，考察它们在不同网络规模下的位置精度、运行时间和内存需求。我们还将所有方法与由后验 Cramér-Rao 下界（PCRLB）给出的理论性能极限进行比较。通过三个不同的场景，我们展示了所提方法的优越性。

I. 引言

位置感知对于各种应用至关重要，例如物联网、自主导航或公共安全[1]–[4]。协同定位（CL）方法旨在估计无线传感器网络中各智能体的位置；智能体可以与邻居通信，并交换有关其位置的信息 [3], [5]–[9]。这既能提高定位精度，也能增强可定位性 [10]，同时避免使用高密度非协作定位所需的锚点部署 [5], [6], [11]–[15]。事实上，利用无线电信号中包含的多径信道信息，可以完全避免锚点基础设施[9]。由于可定位性更高，协同定位比非协作定位更加稳健，因为可以利用网络中的更多信息。这种更强的稳健性对于测量模型信息量很低的场景尤其有用，例如基于接收信号强度的定位 [13], [15]–[17]。协作定位算法具有可扩展性，并且可以通过分布式方式实现，因此特别适用于大规模网络 [18]–[20]。协同定位的另一个关键方面是准确跟踪高维智能体状态。本文提出了一种用于此目的的新方法，可解决以下所述现有最先进算法无法解决的问题。

本研究部分由 Christian Doppler Research Association、奥地利联邦数字与经济事务部以及国家研究、技术与发展基金会资助。

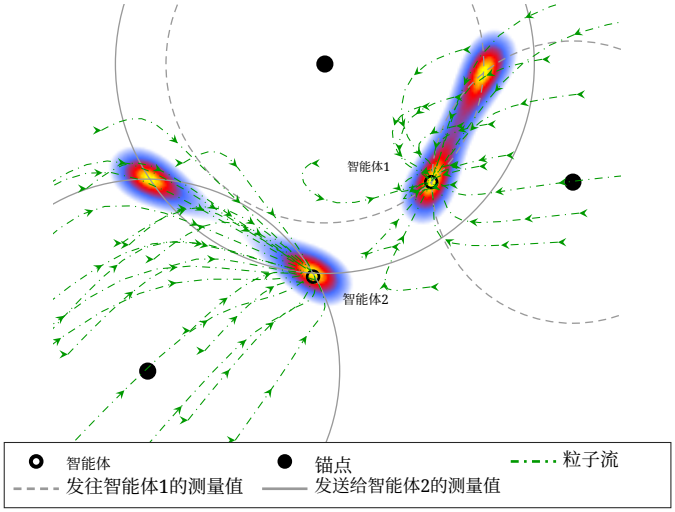


图1：两个协作智能体在三个锚点附近的粒子流可视化结果（绿色点划线）。每个智能体仅与两个锚点（灰色圆圈）相连，这由智能体位置的多模态概率密度函数（颜色映射）表示。

A. 最先进技术

协同定位（CL）的有前景方法基于因子图（FG）框架和消息传递（MP）计算，可分为基于均值场消息传递的方法 [21], [22]和基于置信传播（BP）的方法 [16], [18], [20], [23]–[26]。尤其是，已知基于BP的方法能够通过在循环因子图上执行消息传递，高效地求解高维贝叶斯估计问题，并获得准确结果。和积规则用于计算智能体位置边缘后验概率密度函数（PDF）的近似值（“置信度”） [18], [20]。基于BP的方法具有很强的灵活性，已成功应用于许多不同领域，例如基于无线电信号的同步定位与建图（SLAM） [27]–[29], 、多目标跟踪 [30]–[32], 以及协作多目标跟踪 [33]。基于BP的算法具备出色的可扩展性和分布式特性，使其成为一种

适用于大规模网络协同定位的强大工具 [18]–[20]。基于BP的方法可分为参数化BP算法[25] 和非参数化BP算法 [23]。由于测量模型通常是非线性的，且消息和置信度的计算无法用闭式形式求得，因此通常采用非参数化BP算法，并借助传统的基于自举粒子的实现 [16],[20]。这类方法的一个常见缺点是维数灾难：这是高维基于样本的估计中众所周知的问题，在存在信息性测量时尤为突出。维数灾难可能导致粒子坍缩，也称为粒子退化 [34]。通常只有使用数量庞大到难以实现的粒子，才能避免这一问题并准确表示状态。由于所需内存和计算量都与粒子数量成正比，因此需要开发新的在线估计策略。避免粒子退化的一种常见方法是设计准确的提议分布，或进行正则化 [20], [35], [36]。对于前一种方法，我们必须解决如何设计准确提议分布的问题。此外，正则化必须谨慎处理，因为选择不当可能会引入偏差。

近年来，针对具有高维状态和高度信息丰富的似然模型的非线性系统估计，提出了粒子流（PF） [37]–[42] 。研究表明，由此得到的 PF 粒子滤波器对于非线性估计问题是渐近最优的，即使粒子数量相对较少，也能避免粒子退化。PF 粒子滤波器已成功应用于多传感器定位[43] 和基于 BP 的多目标跟踪 [44] ，其优势在于，与基于自举粒子的实现相比，所需粒子数量显著减少。这些方法的主要缺点是基于联合状态进行估计，导致计算复杂度大幅增加。此外，一些基于粒子流的算法本身复杂度就很高，其计算开销还会随所用粒子数量进一步增加，例如 [43] 中给出的局部 EDH（LEDH）滤波器，或 [40]中描述的随机流。这使得它们不适用于大型网络，也无法实现分布式实现。

B. 论文的贡献与组织结构

本文提出一种混合式基于粒子的PF-BP消息传递算法，用于对具有九维状态（由三维位置、速度和加速度状态向量组成）的移动智能体进行协同定位，并适用于信息量极大的测量模型。在这种场景下，从智能体预测置信分布中抽取样本的自助法粒子BP方法会失效，因为要准确表示智能体的置信分布，需要数量大到不切实际的粒子。我们的方法通过可逆PF [43] 计算BP消息，从而避免粒子退化。可逆PF能够使粒子迁移至概率较高的区域，因此只需相对较少的粒子，就能准确近似BP消息。因此，所提出的算法兼具BP方法的计算效率与可扩展性，以及PF方法的优势。该算法利用了因子分解协同定位问题的结构。这使每次计算所涉及的维度自然减少，同时也缓解了粒子退化问题，并支持分布式实现。例如，图1展示了两个协作智能体的粒子流，它们位于三个锚点附近。由于每个智能体仅与两个锚点相连，智能体位置的概率密度函数呈多模态。纳入协同测量后，粒子流向后验概率密度函数的“正确模态”，从而表示智能体位置的“真实”分布。

数值仿真表明，所提出的PF-BP算法在位置精度方面可以显著优于传统的采用采样-重要性-重采样（简称SIR-BP）的自助式基于粒子的置信传播算法 [20], 、西格玛点置信传播（SP-BP）算法 [25],以及基于粒子的精确Daum-Huang（EDH）滤波器（其堆叠状态向量包含所有智能体状态向量） [43] 。结果表明，所提出的算法是贝叶斯最优的，因为其达到了后验 Cram´ér-Rao 下界（PCRLB） [5],[45],，该下界也可以在等效费 Fisher 信息矩阵的框架下表示 [6]–[8]。与 SIR-BP 算法相比，所提出的算法具有低得多的内存需求，因为在相同位置精度水平下，它所需的粒子数量显著更少。基于粒子的 EDH 滤波器需要针对包含所有智能体状态的堆叠状态向量进行矩阵求逆和矩阵乘法。因此，在粒子数量相同的情况下，所提出算法的内存需求也更低。这是因为采用 PF-BP 时，矩阵求逆和矩阵乘法的运算规模可缩减至联合智能体状态子集的维度。本文的主要贡献可概括如下。

- 我们提出了一种用于动态智能体协同定位的分布式基于粒子的消息传递方法，该方法使用粒子滤波计算BP消息。
- 我们将所提出的PF-BP方法与最先进的协同定位方法进行比较，并展示了其在准确度、运行时间和通信开销方面的优势。
- 我们通过数值结果证明，当智能体可定位时，所提出的用于协同定位的PF-BP方法能够达到PCRLB。
- 我们全面分析了所研究的方法，并着重说明它们在不同场景和应用中的优势。

在本研究中，我们不考虑高斯噪声以外的不确定性，例如漏检、杂波/虚警测量以及测量值的数据关联不确定性 [31], [33], [46], [47]。本文聚焦于动态网络。通过考察统计模型的单个时间步，可以分析静态网络的行为。相较于我们在会议论文 [48] 中对该方法的初步介绍，本文进一步作出以下改进：(i) 在 PF-BP 信念更新方程中也考虑协作邻居智能体的不确定性；(ii) 详细描述所提出的算法；(iii) 将方法扩展至更高的状态维度；(iv) 对成熟的最先进算法以及 PCRLB 理论性能极限进行全面比较，

本文其余部分安排如下。第二节介绍系统与测量模型。第三节阐述问题表述。第四节回顾 PF。第五节介绍消息传递框架并说明所提出的方法。数值实验结果见第六节。第七节总结全文。

符号说明：列向量用粗体小写字母表示，矩阵用粗体大写字母表示。随机变量用无衬线正体表示，其实现值用衬线斜体表示，例如，随机变量 x 和 x 及其相应实现值分别记作 $\boldsymbol{x}$ 和 x。我们将连续随机变量的概率密度函数定义为 $f(\boldsymbol{x})$。对于向量 $\boldsymbol{x}$，其转置记作 $\boldsymbol{x}^{\mathrm{T}}$，欧几里得范数记作 $\|\boldsymbol{x}\|$。向量的均值记作 $\bar{\boldsymbol{x}}$。我们也用此符号表示基于样本的均值和最小均方误差（MMSE）估计。集合 $\mathcal{C}$ 的基数定义为 $|\mathcal{C}|$。此外，我们用符号 $\mathcal{C}\backslash\{i\}$ 表示从集合 $\mathcal{C}$ 中排除元素 $\{i\}$。符号 $\boldsymbol{A}\otimes\boldsymbol{B}$ 表示矩阵 $\boldsymbol{A}$ 与 $\boldsymbol{B}$ 的克罗内克积，而 $\odot$ 表示哈达玛积。$\mathrm{diag}(\cdot)$表示对角矩阵或分块对角矩阵，其主对角线上的元素分别由括号中的元素或矩阵构成。$\boldsymbol{I}_m$ 是维度为 m的单位矩阵。$[\boldsymbol{X}]_{k:l,m:n}$ 表示 $\boldsymbol{X}$ 的一个子矩阵，该子矩阵包含第 k 至 l 行和第 m 至 n 列。符号 $[\boldsymbol{x}]_{k:l}$ 表示 $\boldsymbol{x}$ 的一个子向量，该子向量包含第 k至 l 个元素。时间步 k 用上标（k）表示，而第 u次消息传递迭代记作 $^{[u]}$。 $\nabla_{\boldsymbol{x}}$表示关于 $\boldsymbol{x}^{(k)}$的 Nabla算子。

II. 系统模型

我们考虑一个智能体集合 $\mathcal{C}$ 和一个锚点集合 $\mathcal{A}$。智能体的状态未知，而锚点的状态完全已知。智能体和锚点的数量分别由 $\mathcal{C}$ 和 $\mathcal{A}$的基数表示。我们定义两类测量值：(i) 智能体与锚点之间的测量值 $\mathsf{z}_{i\,a}^{(k)}$，在时间步 k，其中 $i\in\mathcal{C}$ 和 $a\in\mathcal{A}_i^{(k)}$，其中 $\mathcal{A}_i^{(k)}\subseteq\mathcal{A}$ 是对时间 k 的智能体 i 进行测量的锚点集合；(ii) 智能体之间的测量值 $\mathsf{z}_i^{(k)}{}_{,j}$，其中 $i\in\mathcal{C}$ 和 $j\in\mathcal{D}_i^{(k)}$，其中 $\mathcal{D}_i^{(k)}\subseteq\mathcal{C}\backslash\{i\}$ 是在时间 k 与智能体 i 合作的智能体集合。所有时间步的全部测量值组成的堆叠向量记为 $\mathsf{z}=[\mathsf{z}_i^{(1:K)}{}_{,l}\quad]_{i\in\mathcal{C},l\in\mathcal{A}_i^{(1:K)}\cup\mathcal{D}_i^{(1:K)}}$，其中 K 为时间步总数。每个锚点的位置固定不变，不随时间变化。第 i个智能体在时间步 k 的状态表示为 $\mathsf{x}_i^{(k)}=[\mathsf{p}_i^{(k)\mathrm{T}}\ \mathsf{v}_i^{(k)\mathrm{T}}\ \mathsf{a}_i^{(k)\mathrm{T}}]^{\mathrm{T}}\in\mathbb{R}^{9\times1}$ $\mathsf{p}_i^{(k)}\in\mathbb{R}^{3\times1}$ v a，其中 p、$\mathsf{v}_i^{(k)}\in\mathbb{R}^{3\times1}$、$\mathsf{a}_i^{(k)}\in\mathbb{R}^{3\times1}$ 分别为位置、速度和加速度向量。因此，每个智能体状态的维度数为 $N_{\mathrm{D}}=9$。我们将智能体 i在所有时间步的联合状态定义为 $\mathsf{x}_i^{(1:K)}=[\mathsf{x}_i^{(1)\mathrm{T}}\ \ldots\ \mathsf{x}_i^{(K)\mathrm{T}}]^{\mathrm{T}}$。锚点的状态不随时间变化，并假定为已知。第 a个锚点的状态记为 $\mathsf{x}_a=[\mathsf{p}^{x_a}\ \mathsf{p}^{y_a}\ \mathsf{p}_{z_a}]^{\mathrm{T}}\in\mathbb{R}^{3\times1}$。向量 x 表示所有时间步中全部智能体和锚点状态的堆叠向量。其定义为 $\mathsf{x}=[\mathsf{x}_1^{(1:K)\mathrm{T}}\ \ldots\ \mathsf{x}_{|\mathcal{C}|}^{(1:K)\mathrm{T}},\mathsf{x}_{|\mathcal{C}|+1}^{\mathrm{T}}\ \ldots\ \mathsf{x}_{|\mathcal{C}|+|\mathcal{A}|}^{\mathrm{T}}]^{\mathrm{T}}$。

假设第 i个智能体的状态 $\boldsymbol{x}_i^{(k)}$ 按照以下恒加速度模型演化：

$$\boldsymbol{x}_i^{(k)}=\boldsymbol{F}\boldsymbol{x}_i^{(k-1)}+\boldsymbol{G}\boldsymbol{u}^{(k-1)} \tag{1}$$

其中，状态转移矩阵为 $\boldsymbol{F}\in\mathbb{R}^{9\times9}$，矩阵$\boldsymbol{G}\in\mathbb{R}^{9\times3}$ 用于描述状态噪声与状态变量之间的关系。状态噪声向量 $\boldsymbol{u}^{(k)}\in\mathbb{R}^{3\times1}$ 是一个独立同分布（ iid ）的三维高斯随机向量序列，标准差为 σ_a。这些矩阵表示为

$$\boldsymbol{F}=\begin{bmatrix}1&\Delta T&\frac{(\Delta T)^2}{2}\\0&1&\Delta T\\0&0&1\end{bmatrix}\otimes\boldsymbol{I}_3 \tag{2}$$

and

$$\boldsymbol{G}=\begin{bmatrix}\frac{(\Delta T)^2}{2}\\\Delta T\\1\end{bmatrix}\otimes\boldsymbol{I}_3\,. \tag{3}$$

给定运动模型，我们可以定义状态转移概率，并基于常见的统计独立性假设[18], [20]，定义截至时间 K 所有智能体状态的联合先验概率密度函数，如下所示

$$f(\boldsymbol{x}^{(1:K)})=\prod_{k=1}^{K}\prod_{i\in\mathcal{C}}f(\boldsymbol{x}_i^{(0)})f(\boldsymbol{x}_i^{(k)}|\boldsymbol{x}_i^{(k-1)}). \tag{4}$$

截至时间 K 的联合后验概率密度函数表示为

$$f(\boldsymbol{x}^{(1:K)}|\boldsymbol{z}^{(1:K)})\propto f(\boldsymbol{z}^{(1:K)}|\boldsymbol{x}^{(1:K)})f(\boldsymbol{x}^{(1:K)}). \tag{5}$$

假设不同节点和时间步之间的测量值彼此独立 [18], [20], ，我们可以将联合似然函数分解为

$$\begin{aligned}f(\boldsymbol{z}^{(1:K)}|\boldsymbol{x}^{(1:K)})=&\prod_{k=1}^{K}\prod_{i\in\mathcal{C}}\prod_{a\in\mathcal{A}_i^{(k)}}f(z_{i,a}^{(k)}|\boldsymbol{x}_i^{(k)},\boldsymbol{x}_a)\\&\times\prod_{j\in\mathcal{D}_i^{(k)}}f(z_{i,j}^{(k)}|\boldsymbol{x}_i^{(k)},\boldsymbol{x}_j^{(k)}).\end{aligned} \tag{6}$$

将式（4）和式（6）代入式（5），即可通过因子分解形式表示联合后验概率密度函数，结果为

$$\begin{aligned}&f(\boldsymbol{x}^{(1:K)}|\boldsymbol{z}^{(1:K)})\\&\quad\propto\prod_{k=1}^{K}\prod_{i\in\mathcal{C}}f(\boldsymbol{x}_i^{(0)})f(\boldsymbol{x}_i^{(k)}|\boldsymbol{x}_i^{(k-1)})\\&\quad\times\prod_{a\in\mathcal{A}_i^{(k)}}f(z_{i,a}^{(k)}|\boldsymbol{x}_i^{(k)},\boldsymbol{x}_a)\prod_{j\in\mathcal{D}_i^{(k)}}f(z_{i,j}^{(k)}|\boldsymbol{x}_i^{(k)},\boldsymbol{x}_j^{(k)}).\end{aligned} \tag{7}$$

我们利用距离测量值推断智能体的状态。索引分别为 i 和 j 的两个智能体之间，或智能体与锚点之间的测量值为

$$z_{i,j}^{(k)}=h(\boldsymbol{x}_i^{(k)},\boldsymbol{x}_j^{(k)})+n_{i,j}^{(k)} \tag{8}$$

其中 $h(\boldsymbol{x}_i^{(k)},\ \boldsymbol{x}_j^{(k)})=\|\boldsymbol{p}_j^{(k)}-\boldsymbol{p}_i^{(k)}\|$。测量噪声 $\mathsf{n}_{i,\ j}$ 在 i 和 j 上是独立同分布的，服从零均值高斯分布，方差为σ^2。

III. 问题表述

我们的目标是协同估计移动智能体的智能体状态 $\boldsymbol{x}_i^{(k)}$。我们的贝叶斯方法根据截至时间 k 的所有测量值 $\boldsymbol{z}^{(1:k)}$，确定边缘后验概率密度函数 $f(\boldsymbol{x}_i^{(k)}|\boldsymbol{z}^{(1:k)})$。智能体状态 $\boldsymbol{x}_i^{(k)}$ 的估计值由最小均方误差（MMSE）估计器 [49,Ch. 4]给出，即

$$\overline{\boldsymbol{x}}_i^{(k)} = \int \boldsymbol{x}_i^{(k)} f(\boldsymbol{x}_i^{(k)}|\boldsymbol{z}^{(1:k)}) \mathrm{d}\boldsymbol{x}_i^{(k)}. \tag{9}$$

由于式（7）中的联合后验分布通常无法以闭式形式进行直接边缘化，因此通常选择自助法粒子BP [50], [51]实现来近似边缘概率密度函数。当智能体状态维数较高或测量值信息量很大时，这种传统的基于粒子的实现会遭遇粒子退化 [34]。粒子退化会导致智能体信念的表示“错误”，进而损害基于粒子的置信传播算法的收敛性和性能。为解决这一问题，我们提出一种混合PF-BP算法。在介绍所提出的算法之前，先简要回顾PF方法。

IV. 粒子流回顾

对于式（8）中的非线性测量模型，后验概率密度函数 $f(\boldsymbol{x}|\boldsymbol{z}) \propto f(\boldsymbol{z}|\boldsymbol{x})f(\boldsymbol{x})$ 通常用一组加权样本 $\{w^m, \boldsymbol{x}^m\}_{m=1}^M$近似，其中 $\sum_{m=1}^M w^m=1$，样本数量为 M。根据重要性采样原理 [36]，这些样本的计算方式为

$$w^m \propto \frac{f(\boldsymbol{z}|\boldsymbol{x}^m)f(\boldsymbol{x}^m)}{q(\boldsymbol{x}^m|\boldsymbol{z})} \tag{10}$$

采用提议概率密度函数 $q(\boldsymbol{x}|\boldsymbol{z})$，并从中抽取粒子集合 $\{\boldsymbol{x}^m\}_{m=1}^M$。对提议概率密度函数的唯一限制是，它必须与后验概率密度函数具有相同的支撑集，且尾部更重 [52], ，也就是说，它提供的信息较少。除此之外，它可以是任意的。通过选择足够大的M ，重要性采样可以任意精确地近似后验概率密度函数。尽管重要性采样在渐近意义上是最优的，但若未正确选择 $q(\boldsymbol{x}|\boldsymbol{z})$，由于在高维空间中进行正确状态估计所需的粒子数量庞大，通常难以实现。

A. PF 方程的推导

粒子流是一种通过求解偏微分方程 [37], [38], [40], [43], [53], 将粒子从先验概率密度函数迁移到后验概率密度函数的方法。粒子流通过利用同伦性质和福克-普朗克方程（FPE） [54]来描述。FPE 用于求出粒子流，使其根据联合状态 $\mathbf{x}^{(k)}$ 在时刻 k的对数同伦函数，与概率密度函数的流等效。对数同伦函数为[37], [43]

$$\begin{aligned}\log f(\boldsymbol{x}^{(k)};\lambda) = &\log f(\boldsymbol{x}^{(k)}|\boldsymbol{x}^{(k-1)}) \\ &+ \lambda \log f(\boldsymbol{z}^{(k)}|\boldsymbol{x}^{(k)}) - \log Z(\lambda)\end{aligned} \tag{11}$$

其中， $\lambda \in [0,1]$ 是流过程的伪时间，$f(\boldsymbol{x}^{(k)};\lambda)$是流过程中的伪后验分布在时间 λ， $Z(\lambda)$是证据。需要指出的是， $Z(\lambda=0)=1$。对数同伦函数描述了分布的连续平滑变形，该变形从先验概率密度函数 $f(\boldsymbol{x}^{(k)}|\boldsymbol{x}^{(k-1)})$开始，即$\log f(\boldsymbol{x}^{(k)};0) = \log f(\boldsymbol{x}^{(k)}|\boldsymbol{x}^{(k-1)})$，最终得到后验概率密度函数$\log f(\boldsymbol{x}^{(k)};1) \propto \log f(\boldsymbol{x}^{(k)}|\boldsymbol{x}^{(k-1)}) + \log f(\boldsymbol{z}^{(k)}|\boldsymbol{x}^{(k)})$。

假设流遵循如下形式的随机微分方程 [37], [38]

$$d\boldsymbol{x}^{(k)} = \boldsymbol{\zeta}(\boldsymbol{x}^{(k)},\lambda)d\lambda + dw. \tag{12}$$

流方程的详细推导见附录 A。

B. 精确Daum-Huang（EDH）滤波器

该滤波器在每个时间步 k估计$\mathbf{x}^{(k)}$ 的联合智能体状态。我们对其进行回顾，因为它将作为参考方法，也是我们所提出方法的基础性基石。

对于高斯分布 [37]，可以找到式 (39) 中 $\boldsymbol{\zeta}(\boldsymbol{x}^{(k)},\lambda)$ 的解析解，详见附录 A，由此得到 EDH 滤波器 [37], [43]。为满足这些条件，我们将先验概率密度函数近似为高斯分布，其中 $\boldsymbol{R}^{(k)}$和 $\boldsymbol{P}^{(k|k-1)}$ 分别是时间k联合状态的测量噪声协方差矩阵和预测协方差矩阵。根据 EDH 滤波器， $\boldsymbol{\zeta}(\boldsymbol{x}^{(k)},\lambda)$ 的解为 [53]

$$\boldsymbol{\zeta}(\boldsymbol{x}^{(k)},\lambda) = \boldsymbol{A}_\lambda^{(k)}\boldsymbol{x}^{(k)} + \boldsymbol{c}_\lambda^{(k)}. \tag{13}$$

EHD滤波器及其实现的详细说明见附录B，其中还给出了式 (12) 的解。需要指出的是，这种形式的 EDH 只能以集中式方式实现。

V. 消息传递算法与所提方法

在贝叶斯框架下，我们根据边缘后验概率密度函数估计每个智能体的位置。由于通常无法直接对联合后验分布 (7) 进行边缘化，因此我们在表示统计模型的因子图上，按照和积算法规则执行消息传递（MP）。这种所谓的“置信传播（BP）”能够高效地近似得到边缘后验概率密度函数的“置信度” [50], [51]。对于树状图，它能给出精确的边缘概率密度函数；但如果底层因子图中存在环 [50]，则只能进行近似边缘化。在这种情况下，BP消息传递会以迭代方式进行，并且消息的计算顺序有多种选择。我们选择在每次迭代中并行更新所有智能体的置信分布 $i \in \mathcal{C}$。下一节将基于图2中的因子图推导消息传递方案。在第 V-B 节中，我们简要介绍 BP 的标准基于粒子的实现；在第 V-C 节中，则阐述基于相同消息传递方案的所提方法。

A. BP消息传递

根据图2中的因子图，我们定义了用于近似边缘后验概率密度函数的消息传递方案。为便于阅读，使用以下简写

符号：在置信传播的分布式实现中，因子 $f_{ij} \triangleq f(z_{i,j}^{(k)}|\boldsymbol{x}_i^{(k)},\boldsymbol{x}_j^{(k)})$ 表示关于相关智能体 i 和 j 的似然函数，因为在时间 k，节点 $\boldsymbol{x}_i^{(k)}$ 仅能获得测量值 $z_{i,j}^{(k)}$。因此 $f_{ij} \neq f_{ji}$。在集中式实现中，时间 k 智能体 i 和 j 之间的两项测量值均可用。因此，该因子表示为两项测量的似然函数之积，即 $f_{ij} \triangleq f(z_{i,j}^{(k)}|\boldsymbol{x}_i^{(k)},\boldsymbol{x}_j^{(k)})f(z_{j,i}^{(k)}|\boldsymbol{x}_i^{(k)},\boldsymbol{x}_j^{(k)})$，由此得到 $f_{ij}=f_{ji}$。因子 $f_i^{(k)} \triangleq f(\boldsymbol{x}_i^{(k)}|\boldsymbol{x}_i^{(k-1)})$ 对应于状态转移概率密度函数。在时间 $k=0$，它对应于先验概率密度函数 $f(\boldsymbol{x}_i^{(0)})$。因子 $f_{ai} \triangleq f(z_{i,a}|\boldsymbol{x}_i^{(k)},\boldsymbol{x}_a)$ 表示来自锚点测量的信息。由于因子图中存在环，我们采用迭代消息传递方案来近似时间步 k 智能体状态 i 的边缘概率密度函数（置信度）。我们将消息传递迭代 $u \in \{1,\dots,U\}$ 时的置信度定义为所有输入消息的乘积：

$$b^{[u]}(\boldsymbol{x}_i^{(k)}) = \eta(\boldsymbol{x}_i^{(k)}) \prod_{a\in\mathcal{A}_i^{(k)}} \varphi_a(\boldsymbol{x}_i^{(k)}) \prod_{j\in\mathcal{D}_i^{(k)}} \nu_j^{[u-1]}(\boldsymbol{x}_i^{(k)}). \tag{14}$$

消息定义如下：表示智能体 i 状态转移的消息为

$$\eta(\boldsymbol{x}_i^{(k)}) = \int f(\boldsymbol{x}_i^{(k)}|\boldsymbol{x}_i^{(k-1)}) b^{[U]}(\boldsymbol{x}_i^{(k-1)}) d\boldsymbol{x}_i^{(k-1)} \tag{15}$$

而锚点 a 到智能体 i 的消息为

$$\begin{aligned}\varphi_a(\boldsymbol{x}_i^{(k)}) &= \int f(z_{i,a}|\boldsymbol{x}_i^{(k)},\boldsymbol{x}_a)\delta(\boldsymbol{x}_a-\boldsymbol{x}_{\text{true},a})d\boldsymbol{x}_a \\ &= f(z_{i,a}|\boldsymbol{x}_i^{(k)};\boldsymbol{x}_{\text{true},a})\end{aligned} \tag{16}$$

其中，$\boldsymbol{x}_{\text{true},\,a}$ 对应于锚点 a 的真实位置。利用协作智能体 j 提供的外信息 $\psi_i^{[u-1]}(\boldsymbol{x}_j^{(k)})$，协作部分的消息可表示为

$$\nu_j^{[u-1]}(\boldsymbol{x}_i^{(k)}) = \int f(z_{i,j}^{(k)}|\boldsymbol{x}_i^{(k)},\boldsymbol{x}_j^{(k)})\psi_i^{[u-1]}(\boldsymbol{x}_j^{(k)})d\boldsymbol{x}_j^{(k)} \tag{17}$$

对于分布式实现，由于节点 $\boldsymbol{x}_i^{(k)}$ 只能获取测量值 $z_{i,\,j}^{(k)}$，因此可写为。采用集中式方式时，则表示为

$$\begin{aligned}\nu_j^{[u-1]}(\boldsymbol{x}_i^{(k)}) = &\int f(z_{i,j}^{(k)}|\boldsymbol{x}_i^{(k)},\boldsymbol{x}_j^{(k)}) \\ &\times f(z_{j,i}^{(k)}|\boldsymbol{x}_i^{(k)},\boldsymbol{x}_j^{(k)})\psi_i^{[u-1]}(\boldsymbol{x}_j^{(k)})d\boldsymbol{x}_j^{(k)}\end{aligned} \tag{18}$$

因为在时间 k，智能体 i 与 j 之间的两项测量值均可获取。外信息定义为

$$\psi_i^{[u]}(\boldsymbol{x}_j^{(k)}) = \eta(\boldsymbol{x}_j^{(k)}) \prod_{a\in\mathcal{A}_j^{(k)}} \varphi_a(\boldsymbol{x}_j^{(k)}) \prod_{l\in\mathcal{D}_j^{(k)}\backslash\{i\}} \nu_l^{[u-1]}(\boldsymbol{x}_j^{(k)}) \tag{19}$$

其中，符号 $\mathcal{D}_j^{(k)}\backslash\{i\}$ 表示从集合 $\mathcal{D}_j^{(k)}$ 中排除 i。用相应的置信度近似外信息是很常见的做法，由此得到 $\psi_i^{[u]}(\boldsymbol{x}_j^{(k)}) \approx b^{[u]}(\boldsymbol{x}_j^{(k)})$ [18], [20], [24]。这显著降低了计算复杂度，因为无需计算外信息，而外信息在不同协作智能体对之间各不相同。此外，这种做法还能减少

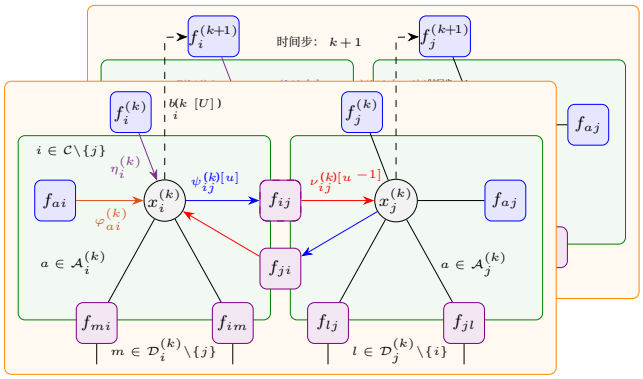


图 2：该图以因子图的形式展示了时间步 k 时系统模型的图形表示。记号 $\mathcal{D}_m^{(k)}\backslash\{l\}$ 表示 $\mathcal{D}_m^{(k)}$ l $b_i^{(k)[U]} \triangleq$ 的所有 $b^{[U]}$ 成员，除了。我们使用简写记号：$(\boldsymbol{x}_i^{(k)})$, $\eta_i^{(k)} \triangleq \eta(\boldsymbol{x}_i^{(k)})$, $\nu_{ji}^{(k)[u-1]} \triangleq \nu_j^{[u-1]}(\boldsymbol{x}_i^{(k)})$, $\varphi_{ai}^{(k)} \triangleq \varphi_a(\boldsymbol{x}_i^{(k)})$ 和 $\psi_{ij}^{(k)[u]} \triangleq \psi_i^{[u]}(\boldsymbol{x}_j^{(k)})$。因子 f_{ij} 会根据采用分布式处理还是集中式处理方案而变化。

智能体之间的通信，因为交换外信息需要点对点通信，而置信度则可以广播 [18], [20], [24]。本文始终采用外信息近似。

智能体的边缘概率密度函数 $f(\boldsymbol{x}_i^{(0:k)}|\boldsymbol{z}^{(1:k)})$ 可由置信度 $b^{[u]}(\boldsymbol{x}_i^{(k)})$ 近似表示，二者仅相差一个归一化常数。我们根据最小均方误差估计器 [49]，估计完成 MP 迭代后第 i 个智能体的状态：

$$\bar{\boldsymbol{x}}_i^{(k)} = \int \boldsymbol{x}_i b^{[U]}(\boldsymbol{x}_i^{(k)}) \mathrm{d}\boldsymbol{x}_i^{(k)}. \tag{20}$$

B. SIR-BP 算法

我们用加权粒子集 $\{w_i^{(k)[u],m}, \boldsymbol{x}_i^{(k)[u],m}\}_{m=1}^M$ 表示 MP 迭代 u 时的置信度。更多详情请参阅 [20]。每次迭代 u 后，我们使用系统重采样 [36]，用一组权重相等的粒子近似第 i 个智能体状态的置信度，如 $\{1/M, \boldsymbol{x}_i^{(k)[u],m}\}_{m=1}^M$ 所示，其中 M 为粒子数。为避免重采样后出现粒子退化，我们可以使用正则化，将重采样粒子集与一个可估计或预先定义的核函数进行卷积 [55]。也就是说，第 m 个粒子 $\acute{\boldsymbol{x}}^{(k)[u]}_{m,\,i}$ 取自高斯分布，其均值为 $\boldsymbol{x}^{(k)[u]}_{m,\,i}$，协方差为 Σ_r。

C. PF-BP 算法

该方法采用与第V-A节所述相同的置信传播消息传递来近似状态的边缘概率密度函数。唯一的区别在于，我们不是对变量节点处的输入消息进行逐点相乘，而是使用粒子流来确定这些消息的乘积。我们用一组权重相等的粒子 $\{1/M, \boldsymbol{x}_i^{(k),m}\}_{m=1}^M$ 表示时刻 k 的智能体状态 i。下面介绍 PF-BP 的基于粒子的实现。

与第IV-B节和附录B相比，第m个粒子的流表示近似边缘后验

智能体 i 在时间步 k、伪时间步 λ_l 和消息传递迭代 u 时的概率密度函数为

$$\boldsymbol{x}_{\lambda_l,i}^{(k)[u],m} = \boldsymbol{x}_{\lambda_{l-1},i}^{(k)[0],m} + \tilde{\boldsymbol{\zeta}}(\boldsymbol{x}_{\lambda_{l-1},i}^{(k)[0],m}, \boldsymbol{x}_{\to i}^{(k)[u-1],m}, \lambda_l)\Delta_l. \quad (21)$$

该递归方程表示对 $a \in \mathcal{A}_i^{(k)}$ 和 $j \in \mathcal{D}_i^{(k)}$ 的输入消息 $\varphi_a(\boldsymbol{x}_i^{(k)})$ 与 $\nu_j^{[u-1]}(\boldsymbol{x}_i^{(k)})$ 进行基于粒子的乘法。消息 $\eta(\boldsymbol{x}_i^{(k)})$ 通过将粒子表示经运动模型传播得到。因此，我们将从提议概率密度函数中抽取的第 m 个粒子定义为 $\boldsymbol{x}_{\lambda_l=0,i}^{(k)[u=0],m} = \boldsymbol{x}_{,i}^{(k|k-1)m}$，其等于运动模型预测的粒子。

变量 $\boldsymbol{x}_{\to i}^{(k)[u]}$ 可视为联合状态，表示在时间 k 对智能体 i 进行测量的智能体的置信度，且该置信度是在消息传递迭代 u 时评估的。$\boldsymbol{x}^{(k)[u]}{}_{,\to i}^{m}$ 表示该联合状态堆叠表示中的第 m 个粒子，后文将对此进行说明。在 $\lambda = 1$ 处，式（21）中表示的粒子与从相应提议密度中抽取的粒子并不完全一致。因此，我们必须使用 [43] 中提到的可逆流，并重新计算粒子的权重。具体做法是根据流结束时（$\lambda = 1$）和开始时（$\lambda = 0$）的粒子表示进行计算，形式如下：

$$w_i^{(k)[u],m} \propto \frac{f(\boldsymbol{x}_{\lambda=1,i}^{(k)[u],m} | \boldsymbol{x}_i^{(k-1),m})}{f(\boldsymbol{x}_{\lambda=0,i}^{(k)[u=0],m} | \boldsymbol{x}_i^{(k-1),m})} \times \prod_{j \in \mathcal{A}_i^{(k)} \cup \mathcal{D}_i^{(k)}} f(z_{i,j} | \boldsymbol{x}_{\lambda=1,i}^{(k)[u],m}, \boldsymbol{x}_j^{(k)[u-1],m}). \quad (22)$$

式 (14) 给出的时刻 k、消息传递迭代 u 下的智能体状态信念 i，由加权粒子集合 $\{w_i^{(k)[u],m}, \boldsymbol{x}_{\lambda=1,i}^{(k)[u],m}\}_{m=1}^M$ 表示。利用加权粒子表示，我们执行系统重采样，以粒子集合近似 $b^{[u]}(\boldsymbol{x}_i^{(k)})$，并为这些粒子赋予均匀权重 $\{1/M, \boldsymbol{x}_i^{(k)[u],m}\}_{m=1}^M$。此处我们再次省略索引 λ，以表示重采样粒子。此时需要指出，消息传递迭代 U 下边缘后验概率密度函数的最终近似记为 $\{1/M, \boldsymbol{x}_i^{(k),m}\}_{m=1}^M$，省略消息传递索引。

我们引入一个新变量 $\boldsymbol{\chi}_i^{(k)[u]}$，表示重采样粒子集。智能体 i 的信念协方差矩阵记为 $\boldsymbol{P}_{(k\,[u])}$。尽管可以使用粒子表示来确定 $\boldsymbol{P}_i^{(k)[u]}$，但我们并未采用这种方法，而是根据下文所述的 UKF 更新步骤来确定它。我们选择这种方法，是因为观察到粒子表示在重采样后可能会退化。

对于每次消息传递迭代 u，我们让智能体状态 i 的粒子流动所有 λ步。此外，我们定义 $\boldsymbol{x}_{\to i}^{(k)[u-1]} = [\boldsymbol{\chi}_j^{(k)[u-1]}]_{j \in \mathcal{D}_i^{(k)}}$，其表示在时间 k 对智能体 i 执行测量的智能体状态，并将其基于样本的均值定义为 $\boldsymbol{x}_{\to i}^{(k)[u-1]}$。协作智能体的状态由其在上一次迭代 $[u-1]$ 时的置信分布表示。此外，我们将消息传递迭代 u 时，智能体 i 在伪时间步 λ_{l-1} 的联合状态及其协作伙伴的堆叠表示定义为 $\boldsymbol{\beta}_{\lambda_{l-1},i}^{(k)[u]} = [\boldsymbol{x}_{\lambda_{l-1},i}^{(k)[0]\mathrm{T}}, \boldsymbol{x}_{\to i}^{(k)[u-1]\mathrm{T}}]^{\mathrm{T}}$，并将其基于样本的均值表示为 $\overline{\boldsymbol{\beta}}_{\lambda_{l-1},i}^{(k)[u]} = [\overline{\boldsymbol{x}}_{\lambda_{l-1},i}^{(k)[0]\mathrm{T}}, \overline{\boldsymbol{x}}_{\to i}^{(k)[u-1]\mathrm{T}}]^{\mathrm{T}}$。由此，我们可以将每个粒子 m 的漂移写为

$$\boldsymbol{\zeta}(\boldsymbol{x}_{\lambda_{l-1},i}^{(k)[0],m}, \boldsymbol{x}_{\to i}^{(k)[u-1],m}, \lambda_l) = \boldsymbol{A}_i \boldsymbol{\beta}_{\lambda_{l-1},i}^{(k)[u],m} + \boldsymbol{c}_i \quad (23)$$

其中 $\boldsymbol{A}_i \triangleq \boldsymbol{A}(\boldsymbol{x}_{\lambda_{l-1},i}^{(k)[0]}, \boldsymbol{x}_{\to i}^{(k)[u-1]}, \lambda_l)$ 和 $\boldsymbol{c}_i \triangleq \boldsymbol{c}(\boldsymbol{x}_{\lambda_{l-1},i}^{(k)[0]}, \hat{\boldsymbol{x}}_{\to i}^{(k)[u-1]}, \lambda_l)$。对于式 (21) 中的流更新，$\tilde{\boldsymbol{\zeta}}(\cdot)$ 由式 (23) 中 $\boldsymbol{\zeta}(\cdot)$ 的前 N_{D} 个元素组成。这对应于智能体状态 i 的边缘分布漂移，因为 $\boldsymbol{x}_i^{(k)}$ 的维度为 N_{D}。智能体状态均值的流与式 (21) 类似，只需将智能体状态的粒子表示替换为式 (43) 中的均值。

据此，我们可以将 $\boldsymbol{A}_i$ 和 $\boldsymbol{c}_i$ 定义为

$$\boldsymbol{A}_i = -\frac{1}{2}\tilde{\boldsymbol{P}}_i \boldsymbol{H}_i^{(k)\mathrm{T}} (\lambda_l \boldsymbol{H}_i^{(k)} \tilde{\boldsymbol{P}}_i \boldsymbol{H}_i^{(k)\mathrm{T}} + \boldsymbol{R}_i^{(k)})^{-1} \boldsymbol{H}_i^{(k)} \quad (24)$$

$$\boldsymbol{c}_i = (\boldsymbol{I}_{N_{\mathrm{D}}(|\mathcal{D}_i^{(k)}|+1)} + 2\lambda_l \boldsymbol{A}_i) \Big[(\boldsymbol{I}_{N_{\mathrm{D}}(|\mathcal{D}_i^{(k)}|+1)} + \lambda_l \boldsymbol{A}_i) \times \tilde{\boldsymbol{P}}_i \boldsymbol{H}_i^{(k)\mathrm{T}} (\boldsymbol{R}_i^{(k)})^{-1} (\boldsymbol{z}_i - \boldsymbol{\nu}_i) + \boldsymbol{A}_i \overline{\boldsymbol{\beta}}_{\lambda=0,i}^{(k)[u]} \Big] \quad (25)$$

with

$$\boldsymbol{\nu}_i = [h(\overline{\boldsymbol{x}}_{\lambda_{l-1},i}^{(k)[0]}, \overline{\boldsymbol{\vartheta}}_q^{(k)})]_{q \in \mathcal{A}_i^{(k)} \cup \mathcal{D}_i^{(k)}} - \boldsymbol{H}_i^{(k)} \overline{\boldsymbol{\beta}}_{\lambda_{l-1},i}^{(k)[u]} \quad (26)$$

其中，$\boldsymbol{\nu}_i$ 对应于线性化导致的模型失配，而 $\boldsymbol{\vartheta}^{(k)} = [\boldsymbol{x}_{\mathrm{true}}{}^{\mathrm{T}}{}_{,\mathcal{A}_i(1)} \cdots, \boldsymbol{x}_{\mathrm{true},\,\mathcal{A}_i(|\mathcal{A}_i|)}^{\mathrm{T}} \boldsymbol{x}_{\to i}^{(k)[u-1]\mathrm{T}}]^{\mathrm{T}}$、$\overline{\boldsymbol{z}}_i = [z_{i,j}]_{j \in \mathcal{A}_i^{(k)} \cup \mathcal{D}_i^{(k)}}$ 和 $\boldsymbol{x}_{\lambda_l}^{(k)[u]}{}_{,\,\overline{i}} = (1/M) \sum_{m=1}^M \boldsymbol{x}_i^{(k)[u]}{}_{,\lambda_l}$。下文将定义所有其他涉及的向量和矩阵。

观测矩阵 $\boldsymbol{H}_i^{(k)}$ 的维度为 $|\mathcal{A}_i^{(k)}| + |\mathcal{D}_i^{(k)}| \times N_{\mathrm{D}}\ 1 + |\mathcal{D}_i^{(k)}|$（）（），这等价于智能体 i 的测量值数量乘以所有相关状态维度之和。$\boldsymbol{H}_i^{(k)}$ 由 N_{D}维元素组成

$$[\boldsymbol{H}_i^{(k)}]_{\tilde{o},N_{\mathrm{D}}\tilde{p}-N_{\mathrm{D}}+1:N_{\mathrm{D}}\tilde{p}} = \frac{\partial h(\boldsymbol{x}_p^{(k)}, \boldsymbol{x}_o^{(k)})}{\partial \boldsymbol{x}_p^{(k)}} \bigg|_{\boldsymbol{x}_p^{(k)} = \hat{\overline{\boldsymbol{\beta}}}_{\tilde{p}}} \quad (27)$$

对于 $p \in \{i\} \cup \mathcal{D}_i$，它是一个索引为 $\tilde{p}$的有序集合，表示协作伙伴的索引；有序集合 $o \in \mathcal{A}_i^{(k)} \cup \mathcal{D}_i^{(k)}$ 的索引为 $\tilde{o}$，用于确定第 o个测量值的索引。导数在 $\hat{\boldsymbol{\beta}}_{\tilde{p}} = [\boldsymbol{\beta}_{\lambda_{l-1},i}^{(k)[u]}]_{N_{\mathrm{D}}\tilde{p}-N_{\mathrm{D}}+1:\ N_{\mathrm{D}}\tilde{p}}$处求值。

式（27）中的前三个元素对应于相对于位置坐标的导数。接下来的三个元素对应于相对于速度坐标的导数，最后三个元素对应于相对于加速度坐标的导数。包含相对于速度和加速度的导数的元素均为零，因为观测模型仅取决于位置。

测量噪声协方差矩阵 $\boldsymbol{R}_i^{(k)}$的维度为（$|\mathcal{A}_i^{(k)}| + |\mathcal{D}_i^{(k)}|$）×（$|\mathcal{A}_i^{(k)}| + |\mathcal{D}_i^{(k)}|$），其主对角线上为 σ^2，其余位置均为零。我们还将时间 k时所涉及状态的分块对角协方差矩阵定义为

$$\tilde{\boldsymbol{P}}_i = \mathrm{diag}\left(\boldsymbol{P}_i^{(k|k-1)}, \ldots, \boldsymbol{P}_m^{(k)[u-1]}, \ldots\right) \quad (28)$$

算法1 所提出的PF-BP算法

对于 $i = 1 : |\mathcal{C}|$，**执行**

使用均值初始化高斯先验分布 $\bar{\boldsymbol{x}}_i^{(0)}$ 和协方差矩阵 $\boldsymbol{P}_i^{(0)}$.

从先验分布中抽取粒子 $\{1/M, \boldsymbol{x}_i^{(0),m}\}_{m=1}^{M}$ 分布

</style结束

对k =1:K 执行

对 $i = 1 : |\mathcal{C}|$ **执行**

根据粒子和协方差矩阵进行预测 (1) 和 (29) 预测粒子和协方差矩阵。

确定基于样本的均值 $\bar{\boldsymbol{x}}_{\lambda=0,i}^{(k)[0]}$

结束循环

对 $u = 1 : U$ **执行**

对 $i = 1 : |\mathcal{C}|$ **执行**

根据 (21) 计算流 对所有 λ 步执行（使用 (23)–(28)）

根据 (22) 重采样粒子，以获得 $\{1/M, \boldsymbol{x}_i^{(k)[u],m}\}_{m=1}^{M}$

确定基于样本的均值 $\bar{\boldsymbol{x}}_i^{(k)[u]}$

计算 $\boldsymbol{P}_i^{(k)[u]}$ 根据（30），在 $\boldsymbol{x}_i^{(k)[u]}$

可选：对重采样粒子进行正则化以及 $\boldsymbol{P}_i^{(k)[u]}$ 根据 (33)

结束循环

结束循环

对于 $i = 1 : |\mathcal{C}|$，**执行**

根据以下内容确定最小均方误差估计 基于样本的均值 $\bar{\boldsymbol{x}}_i^{(k)[U]}$

结束循环

结束循环

其中，$\boldsymbol{P}_i^{(k|k-1)}$ 是智能体状态 i 的预测协方差矩阵，而 $\boldsymbol{P}_m^{(k)[u-1]}$是所有其他相连智能体 $m \in \mathcal{D}_i^{(k)}$ 的状态协方差矩阵，这些矩阵是在上一轮消息传递迭代 $u-1$的流时间 $\lambda = 1$ 确定的。与 [43], 类似，这些协方差矩阵分别通过无迹卡尔曼滤波协方差矩阵预测与更新计算，即

$$\boldsymbol{P}_i^{(k|k-1)} = \boldsymbol{F}\boldsymbol{P}_i^{(k-1)[U]}\boldsymbol{F}^{\mathrm{T}} + \boldsymbol{Q} \tag{29}$$

$$\boldsymbol{P}_i^{(k)[u]} = \boldsymbol{P}_i^{(k|k-1)} - \tilde{\boldsymbol{K}}^{[u]}\tilde{\boldsymbol{P}}_{zz}\tilde{\boldsymbol{K}}^{[u]\mathrm{T}} \tag{30}$$

其中，$\tilde{\boldsymbol{K}}^{[u]}$ 再次表示消息传递迭代u 时的卡尔曼增益，因为它取决于相关智能体状态的置信分布，~而 $\boldsymbol{P}_{zz}$ 是测量协方差矩阵。如上所述，我们在每次消息传递迭代结束时执行系统重采样，从而得到 $\{1/M, \boldsymbol{x}_i^{(k)[u],m}\}_{m=1}^{M}$。注意，协方差矩阵 $\boldsymbol{P}_i^{(k)[u]}$ 是在基于样本的均值 $\boldsymbol{x}_i^{(k)[u]}$ 处计算的。除粒子外，我们还用均值和协方差矩阵表示智能体 i 在时刻 k 、消息传递迭代 u时的边缘后验概率密度函数。在消息传递迭代 U时，我们根据各智能体状态的基于样本的均值，确定每个智能体状态的最小均方误差估计。我们采用 λ 指数间隔，如 [38],所建议；在我们的仿真中，与步数相同的线性间隔相比，这能得到更准确的位置估计。算法1总结了PF-BP的基于粒子的实现。

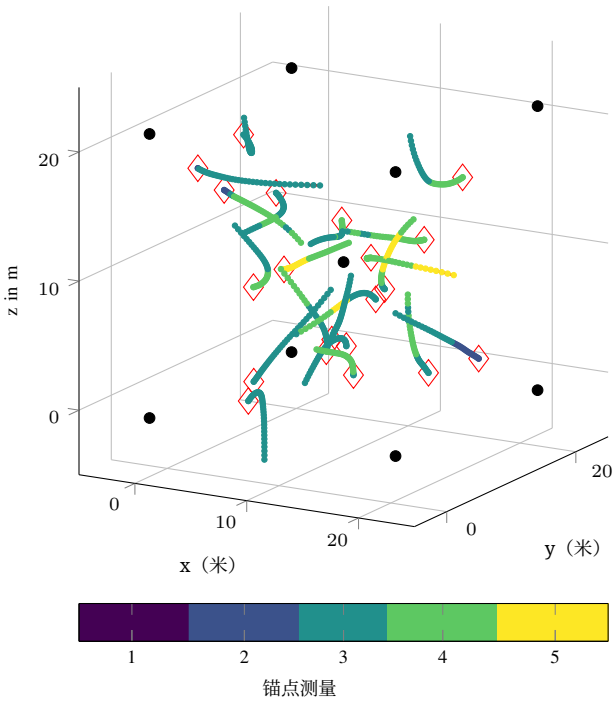


图3：20 个智能体的轨迹示例。锚点以黑色表示。智能体的初始位置用红色菱形标记，轨迹用红色表示。彩色散点表示智能体沿轨迹移动时与锚点的连接数量。通信范围为 $r_{\max} = 18$ m。每个时间步，智能体至少与一个锚点保持连接。

VI. 算法评估

在本节中，我们针对不同网络规模和连通性，评估所提出的基于动态网络的算法。我们采用式（1）给出的九维恒加速度运动模型（由三维位置、速度和加速度状态向量组成）。我们将其性能与第五节 B 部分所述的基于自举粒子的BP算法（称为SIR-BP）、SP-BP算法 [25],以及完全联合的基于粒子的EDH滤波器 [43]进行比较。此外，我们还展示相对于PCRLB [5], [45]的理论性能极限。我们通过位置（RMSE$_{\mathrm{P}}$）、速度（RMSE$_{\mathrm{V}}$）和加速度（RMSE$_{\mathrm{a}}$）最小均方误差估计的均方根误差（RMSE）、位置误差的累积频率（CF）以及每个时间步的运行时间来确定性能。此外，我们还展示位置误差超过位置误差阈值的中断概率。中断定义为位置误差高于该阈值。测量模型的不确定性为 $\sigma = 0.1$ m。在以下仿真中，我们使用9个锚点，并采用两种不同数量的智能体，分别定义为 $N_{\mathrm{agent}} \in \{5, 20\}$。每次试验中，智能体的真实位置均匀分布在一个尺寸为20m $\times$ 20m $\times$ 20m的空间内。每个智能体的真实速度初始化为指向场景中心的单位向量，而真实加速度初始化为零。根据式（1）所示的恒加速度模型，在三维空间中生成智能体轨迹，其中$\Delta T = 0.1$ s，且 $\boldsymbol{u}^{(k)}$ 的标准差为 σ_a =0.15 m/s^2。位置的先验分布（除SIR-BP算法外），以及每个智能体状态的速度和加速度先验分布为（

SIR-BP算法除外），每个智能体状态$\mathbf{x}_i$的位置、速度和加速度均初始化为高斯分布，其均值为$\boldsymbol{x}_i^{(0)} = [\boldsymbol{p}_i^{(0)\mathrm{T}} \boldsymbol{v}_i^{(0)\mathrm{T}} \boldsymbol{a}_i^{(0)\mathrm{T}}]^{\mathrm{T}}$（稍后给出），协方差矩阵为

$$\boldsymbol{P}_i^{(0)} = \mathrm{diag}([(\boldsymbol{\sigma}_p^2)^{\mathrm{T}}, \Delta T^2(\boldsymbol{\sigma}_{a_{\mathrm{init}}}^2)^{\mathrm{T}}, (\boldsymbol{\sigma}_{a_{\mathrm{init}}}^2)^{\mathrm{T}}]) \tag{31}$$

其中 $\boldsymbol{\sigma}_p^2 = [\sigma_{px}^2, \sigma_{py}^2, \sigma_{pz}^2]^{\mathrm{T}}$。我们将位置在所有维度上的先验标准差设为相同，并取 20 m。对于 $\boldsymbol{\sigma}_{a_{\mathrm{init}}}^2$，我们也将其在所有维度上设为相同，其值为 $\boldsymbol{\sigma}_{a_{\mathrm{init}}}^2 = [\ 10\sigma_{\boldsymbol{v}_i^{(0)}}{}^2,\ 10\sigma_{\boldsymbol{a}_i^{(0)}}{}^2,\ 10\sigma_{a}{}^2]^{\mathrm{T}}$ ( ) ( ) ( )。分别对应速度和加速度的均值，取自由式 (31) 中协方差矩阵定义的零均值高斯分布。对应位置的均值 $\boldsymbol{p}_i^{(0)}$ 则从支撑体积中均匀抽取。对于 SIR-BP 算法，表示位置的粒子从支撑体积中均匀抽取。相比之下，对于 EDH 滤波器和 PF-BP 算法，粒子从高斯先验分布中抽取。图 3 给出了一个包含 20 个智能体且通信范围为 $r_{\max}$ = 18 m 的动态场景实例。该图还显示了锚点在支撑体积各角落的布置，以及单个锚点在中心的布置。此外，我们用颜色标示了智能体在其轨迹各点处测得的锚点数量。该设置确保每个智能体在每个时间步都至少处于一个锚点的通信范围内。若要仅根据锚点测量实现智能体的完全定位，需要来自四个不同锚点的测量，且这些锚点的位置不能共面。正如图 3 所示，对于大多数轨迹，若没有协作测量，智能体将无法定位。

我们模拟智能体的 200 条轨迹，共运行 $K = 40$ 个时间步。对于基于PF的算法，我们采用 20 λ 个时间步和 200 个粒子；对于 SIR-BP 算法，则采用 100000 个粒子。作为额外的基准，我们为 SIR-BP 算法采用 1000000 个粒子，并将其记为 SIR-BP$_{\mathrm{Mil}}$。我们将 MP迭代次数固定为 2。更多迭代会耗费更多时间，而且对基于BP的算法收敛特性的改善微乎其微。本节后文将进一步探讨这一问题。由于通常会采用正则化来避免粒子退化 [55], ，我们研究了正则化对所有所述方法的影响。为此，我们对所有考察的算法中的速度和加速度进行正则化，分别采用 $\sigma_{r_{\mathrm{vel}}} = 0.15$ m/s 和$\sigma_{r_{\mathrm{acc}}} = 0.15$ m/s^2 。具体做法如下：我们定义一个协方差矩阵为

$$\boldsymbol{\Sigma}_r = \mathrm{diag}([0, 0, 0, \sigma_{r_{\mathrm{vel}}}^2, \sigma_{r_{\mathrm{vel}}}^2, \sigma_{r_{\mathrm{vel}}}^2, \sigma_{r_{\mathrm{acc}}}^2, \sigma_{r_{\mathrm{acc}}}^2, \sigma_{r_{\mathrm{acc}}}^2]). \tag{32}$$

对于 UKF 更新和 SP-BP，我们将该协方差加到每个边缘状态的估计协方差中。例如，使用式 (30) 可得

$$\boldsymbol{P}_i^{(k)[u]} = \boldsymbol{P}_i^{(k|k-1)} - \tilde{\boldsymbol{K}}^{[u]} \tilde{\boldsymbol{P}}_{zz} \tilde{\boldsymbol{K}}^{[u]\mathrm{T}} + \boldsymbol{\Sigma}_r. \tag{33}$$

对于基于粒子的方法，重采样后，我们为每个粒子抽取 $\boldsymbol{x}_{i}^{(k),m}$ 一个新粒子 $\acute{\boldsymbol{x}}_{i}^{(k),m}$，其服从均值为 $\boldsymbol{x}_{i}^{(k),m}$ 和协方差为 $\boldsymbol{\Sigma}_r$的高斯分布。以下图中采用正则化的结果以虚线或点线表示，图例中则标注为“reg”。

表 I：5 个智能体在联合处理和分布式处理（分布）下的结果的每个时间步运行时间。对于分布式处理，结果以每个智能体的运行时间表示。

	$r_{\max}$	SP-BP	EDH	PF-BP	SIR-BP	SIR-BP$_{\mathrm{Mil}}$
联合	18	3 ms	10 毫秒	50 毫秒	0.44 秒	4.4 秒
	∞	4 ms	20 毫秒	60 毫秒	0.51 秒	5.1 秒
分布	18	0.6 毫秒	-	10 毫秒	0.09 秒	0.9 秒
	∞	0.8 毫秒	-	12 毫秒	0.10 秒	1.2 秒

1）场景 *I*：我们评估了一个包含5个智能体、通信范围各不相同的场景 $r_{\max}$。当通信范围为 $r_{\max}$ = 18 m时，每个智能体至少与一个锚点连接，这与图 3 中给出的场景类似。该设置的结果见图 4a-4d，其中展示了整体轨迹的累积分布函数（CF），以及每个时间步的位置、速度和加速度的均方根误差。显然，在不进行正则化的情况下，EDH 滤波器和所提出的PF-BP算法在准确度方面显著优于 SP-BP 算法和SIR-BP 算法。表 I 展示了各算法在联合处理和分布式处理下每个时间步的运行时间。对于分布式处理，运行时间按每个智能体、每个时间步计算。对于智能体数量较少且所选粒子数量有限的情况，SP-BP 算法的运行时间优于所有其他方法。

在最初几个时间步中，智能体状态的部分边缘后验概率密度函数仍呈多模态，可由SIR-BP 算法的粒子很好地表示。因此，SIR-BP 算法能够更快地收敛到后验概率密度函数的“正确模态”，从而使智能体轨迹起始阶段的位置误差显著降低（见图 4b）。然而，几步之后可以观察到，在几乎每次仿真运行中，SIR-BP 算法都会发散，因为所选粒子数量（100 000）仍然太少，无法充分表示9维智能体状态向量。采用正则化后，SIR-BP 算法的性能大幅提升。不过，我们仍能观察到均方根误差中存在显著偏差，这表明所选粒子数量仍然过少。使用1 000000个粒子并进行正则化后，SIR-BP 算法的性能几乎达到PCRLB水平；但代价是运行时间显著增加（见表 I），因此不适用于实时应用和存在内存限制的系统。通过使用更多粒子可以消除出现的微小偏差（未显示）。SP-BP 算法也受益于正则化，因为正则化可使最小均方误差估计随时间更快地收敛至PCRLB。不过，与PCRLB相比，其可达到的准确度仍然很低。此外，观察发现，SP-BP 给出的后验协方差矩阵存在显著的过度自信（未显示）。对于两种基于粒子滤波的方法，正则化的影响都很小。

对于全连接的智能体网络（测量模型信息量很高），从图 4e–4h 中可以清楚地看出，基于粒子滤波的两种方法都具有优势。与其他方法相比，所提出的 PF-BP 算法能快得多地达到理论性能极限。EDH 滤波器达到

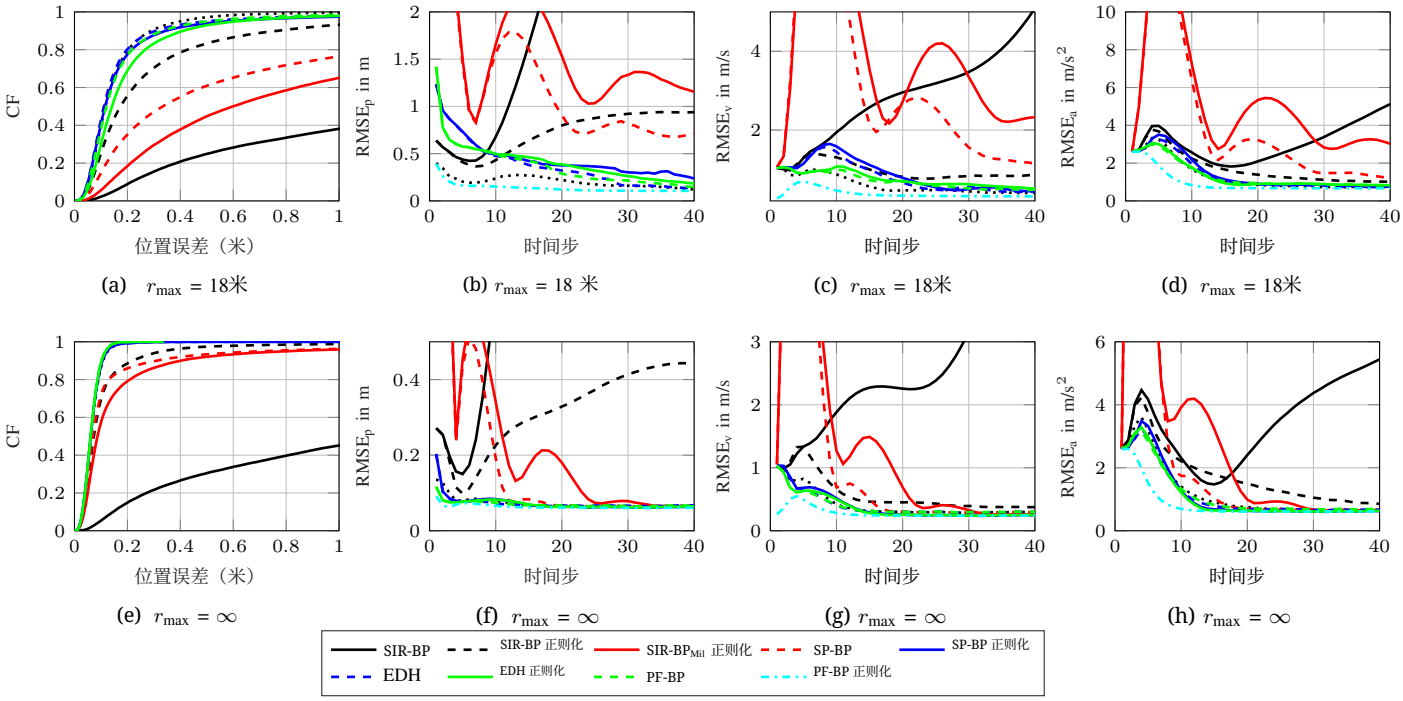


图 4：通信范围 r_{max} 对 5 个智能体的准确度性能的影响；200 次仿真运行中，$\sigma = 0.1$ m。我们展示了整个轨迹上的位置误差累积分布函数，以及每个时间步的智能体状态均方根误差，并分别考察位置、速度和加速度。理论性能极限以 PCRLB 表示。正则化以 reg 表示。

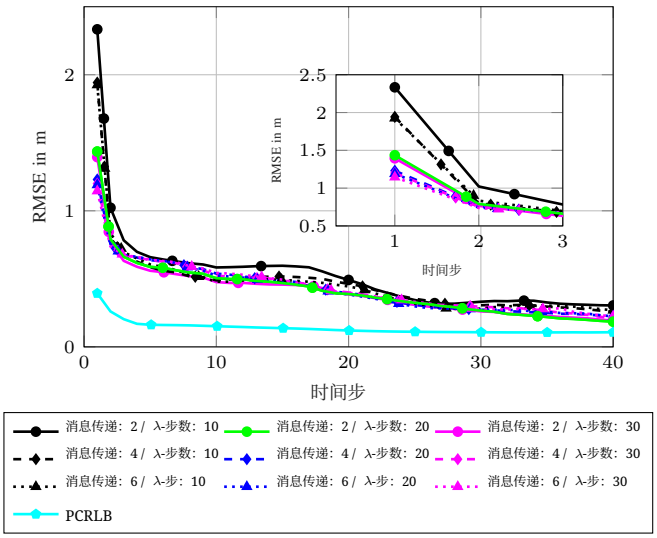


图 5：PF-BP 随消息传递迭代次数和伪时间步变化的收敛行为。结果取 200 次仿真运行的平均值。所有其他仿真均采用与绿色曲线对应的设置。

PCRLB 只需几个时间步。SP-BP 算法则需要显著更多的时间步才能收敛到 PCRLB。使用 100 000 个粒子时，SIR-BP 算法在每次仿真运行中，无论是否进行正则化，显然都会发散。即使使用 1 000 000 个粒子，SIR-BP 算法也只有在启用正则化时才能收敛。图 4f 显示，在这种情况下，SIR-BP 算法也能达到位置 PCRLB；然而，由于正则化，速度和加速度的均方根误差存在偏差。由于运行时间长且内存需求巨大，我们不展示使用更多粒子的结果。

两种基于粒子滤波的方法无需正则化即可达到 PCRLB。图 4g 显示，对于测量信息量很大的测量模型，对基于粒子滤波的方法进行正则化只会使所有状态产生误差偏差，适得其反。图 4g 还表明，SP-BP 和 SIR-BP 算法可以从正则化中受益，因为若不进行正则化，它们对速度和加速度的估计需要更多时间步才能收敛，甚至会发散。我们得出结论，正则化应谨慎使用，因为它会显著影响误差偏差。

表 I 汇报了两种智能体网络中所研究算法的运行时间。这些时间分别基于集中式处理和分布式处理测得。结果表明，尽管集中式处理时 PF-BP 的计算时间比 EDH 滤波器更长，但在分布式处理中，每个智能体的计算时间更短或相近。

此外，我们还研究了所提出方法关于 $r_{max} = 18$ m 的收敛行为。图 5 展示了不同消息传递迭代次数和不同 λ 步数下，轨迹随时间步向 PCRLB 收敛的情况。可以看出，增加 λ 步数总是比增加消息传递迭代次数更有利。因此，如本节开头所述，所有仿真中消息传递迭代次数均固定为 2，λ 步数固定为 20。采用这组参数得到的结果以绿色标示。

此外，我们在图 6 中展示了位置误差的中断概率

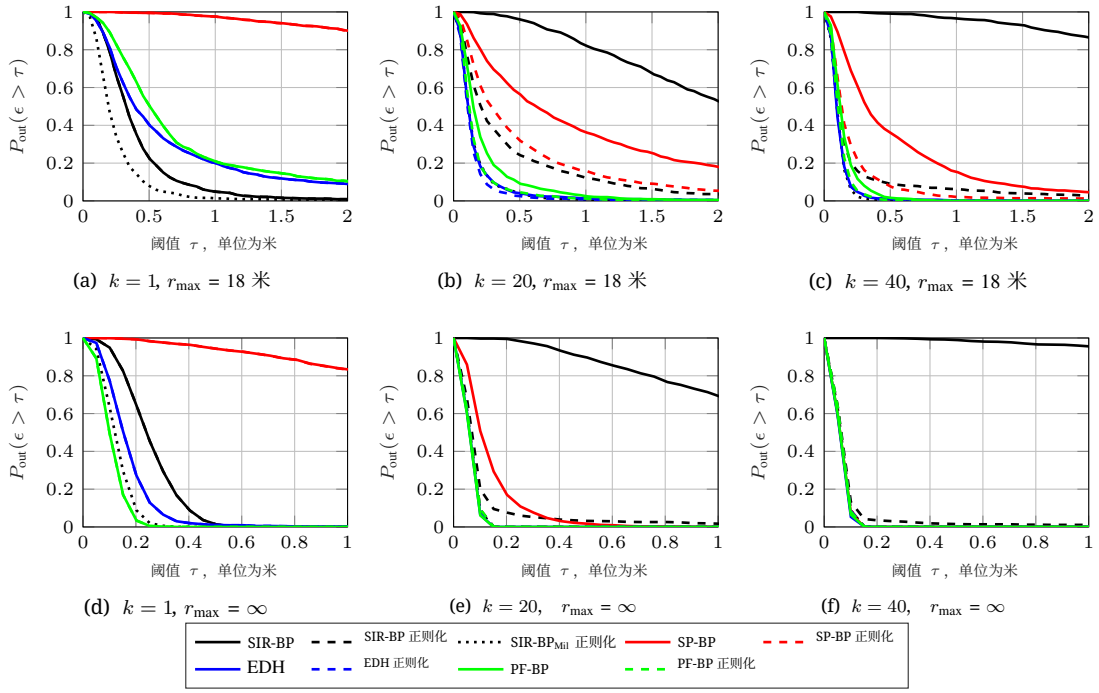


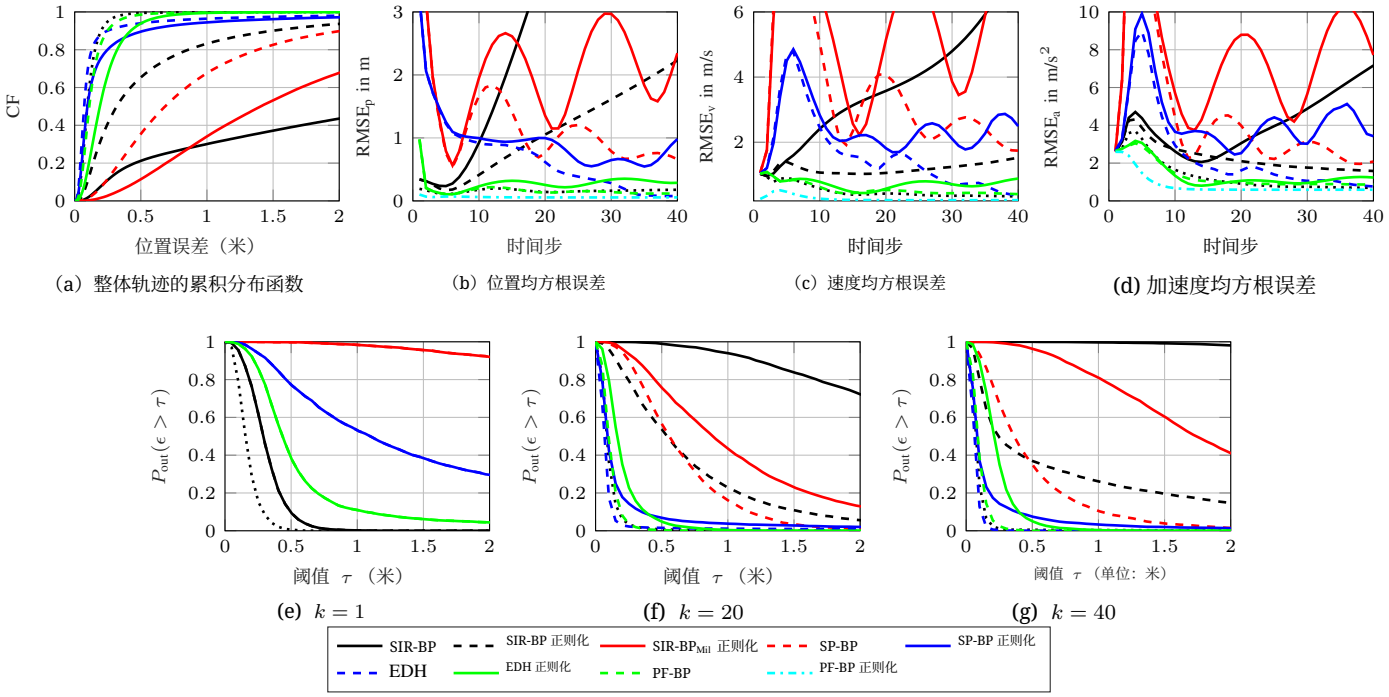
图 6：五个智能体场景下，所研究算法的位置误差中断概率。第一行显示通信范围为 $r_{\max}$=18 m 时的中断概率，第二行则显示全连接情况下的中断概率。该概率在特定时间步 k进行评估。正则化以 reg 表示。

表 II：20 个智能体的结果中，联合处理与分布式（distr.）处理的每个时间步的运行时间。对于分布式处理，结果以每个智能体的运行时间给出。

	SP-BP	EDH	PF-BP	SIR-BP	SIR-BP$_{\text{Mil}}$
联合	0.07 秒	0.25 秒	0.9 秒	3.6 秒	40 s
分布	0.004 秒	-	0.05 秒	0.18 秒	2 s

$P_{\text{out}}(\epsilon > \tau)$，其中 ϵ 是位置误差，τ 是以米为单位的位置阈值。我们在三个时间步 $k \in \{1, 20, 40\}$ 对其进行评估。在 $k = 1$ 时，可以看到不同算法的优势。图 6a 显示，在 $k = 1$ 时，对于 $r_{\max} = 18$ 米的情况，采用 1 000 000 个粒子的 SIR-BP 算法结果最准确，其次是采用 100 000 个粒子的 SIR-BP。这是因为第一步中并非每个智能体都可定位，而且如上所述，只要粒子数量足够，SIR-BP 就能表示任意概率密度函数。在图 6d 中，由于场景是全连接的，位置状态不存在多模态性。因此，PF-BP 算法的单模态近似足以正确表示智能体状态。因此，它的准确度高于采用 1 000 000 个粒子的 SIR-BP。对于 $k = 20$，除未进行正则化的 SIR-BP 算法外，所有基于粒子的方法表现相近。图 6b 中 SP-BP 的估计值仍有偏差，而在图 6e 中则接近最优结果。在最后一步，我们看到，若算法收敛，则所有算法的表现大致相同；这与图 4f 中的结果一致，其中所有受考察的方法在最后一个时间步均达到 PCRLB。

2) 场景 II：图 7 展示了 20 个智能体在通信范围为 $r_{\max} = 18$ 米时的结果。结果与图 4 所示结果相似，但有两个主要区别。首先，我们可以观察到，没有任何一个在所定义的参数化下，所研究的方法均达到了 PCRLB。然而，PF-BP 的偏差最小。此外，我们可以看到，在 $k = 1$ 处，基于粒子滤波的方法所得的估计值存在显著差异。由于此时联合状态有 180 个维度，而五个智能体场景中的联合状态只有 45 个维度，因此 EDH 滤波器在正确表示状态方面遇到了更多困难。PF-BP 算法确定各智能体的边缘后验概率密度函数，并且仅基于联合状态的一个子集计算流，即智能体 i 的状态以及与其相连的所有其他智能体的状态。因此，状态维度要小得多，这也减轻了粒子退化的影响。在相同的参数设置下，这使其结果与图 4b 中五个智能体场景的结果相近。在 $k = 1$ 处与 SIR-BP 算法的差异表明，PF-BP 算法无法解析多模态性。我们可以观察到，在此场景和特定参数设置下，所有研究方法均受益于正则化。图 7b 中 PF-BP 算法的均方根误差在未进行正则化时存在恒定偏差。增加粒子数量可以解决这一问题，但会延长运行时间。EDH 滤波器也存在同样的情况。我们还可以看到，基于粒子滤波的方法是唯一能够在整个轨迹的时间范围内，以合理的计算时间达到 PCRLB 的方法。表 II 总结了联合处理和分布式处理下每个时间步的运行时间。可以看出，SIR-BP 算法运行时间较长，因此不适用于实时应用。PF-BP 算法的运行时间也长于 EDH 滤波器，但仅在联合处理时如此，因此它适用于实时应用。SP-BP 的运行时间优于所有其他方法，但完全无法收敛至估计的理论极限



（a）整体轨迹的累积分布函数　（b）位置均方根误差　（c）速度均方根误差　(d) 加速度均方根误差

(e) $k = 1$　(f) $k = 20$　(g) $k = 40$

图 7：在包含 20 个智能体且 $r_{\max} = 18$ m 的场景中，所研究算法的准确度性能可视化。第一行展示了整个轨迹上的位置误差累积分布函数，以及每个时间步的智能体状态均方根误差，分别考察位置、速度和加速度。第二行展示了特定时间步 k 上位置误差的中断概率。结果基于 $\sigma = 0.1$ m 和 200 次仿真运行。正则化以 reg 表示。

性能。

需要注意的是，对于时间 $k = 1$ 智能体状态的信息量很高的先验分布，基于粒子滤波的方法仍会比 SIR-BP 和 SP-BP 算法具有更高的准确度。不过，尤其对于 SP-BP 算法而言，差异要小得多。

下文将总结对比方法和所提出算法的优缺点。

- SIR-BP 算法需要大量粒子，才能正确表示 9 维智能体状态的后验概率密度函数。因此，该算法运行时间较长，且需要大量内存。然而，随着粒子数量增加，SIR-BP 算法有望渐近地正确表示智能体状态的后验概率密度函数。因此，它能够捕捉后验概率密度函数中的多模态性。
- SP-BP 算法的计算需求较低，因此运行时间较短。然而，对于高维智能体状态，随着时间推移，其均方根误差收敛至较小值的速度较慢。
- 基于粒子的 EDH 滤波器适用于小型智能体网络，因为它能够提供达到 PCRLB 水平的位置精度，且运行时间较短。然而，对于较大的网络，最小均方误差估计随时间收敛的速度相对较慢，也就是说，需要经过许多时间步才能达到 PCRLB 水平。由于采用联合状态表示，该滤波器在智能体数量增加时也难以扩展。
- 所提出的 PF-BP 算法能够以 PCRLB 水平实现高位置精度，同时运行时间较短每个时间步的分布式处理耗时较短。由于能够进行分布式实现，该算法还可以快速收敛，并能很好地扩展到更多智能体。

关于通信开销，我们可以得出以下结论：SP-BP 和 PF-BP 使用高斯近似，这意味着智能体状态由高斯分布表示。因此，每个智能体只需传输与其置信度对应的均值和协方差，而不像 SIR-BP 那样传输所有粒子。对于 PF-BP，每个智能体都必须从该高斯分布中进行本地采样，以便在测量更新步骤中执行粒子流过程。EDH 无法以分布式方式实现，因此必须由中央计算单元收集所有测量值并进行计算。

为了更清楚地展现所提方法的优势，我们测定了所研究算法在集中式和分布式处理下的运行时间。结果表明，尽管集中式处理时 PF-BP 的计算时间比 EDH 滤波器更长，但在分布式处理时，各智能体的计算时间更短或与之相近。

VII. 结论

我们提出了一种基于置信传播（BP）和粒子流的贝叶斯方法，用于协作定位与导航。我们的方法尤其适用于智能体状态维度较高、且测量模型具有丰富信息的非线性场景。为避免此类场景中的粒子退化，我们采用可逆PF计算BP消息。因此，在具有9维智能体状态和距离测量值的协同定位场景中，所提出的PF-BP算法能够达到PCRLB级别的位置精度

。数值结果表明，与传统SIR-BP算法以及应用于协同定位的基于粒子的EDH滤波器相比，该方法的计算需求和内存需求均有所降低。此外，与SIR-BP相比，其通信开销显著降低，且与依赖类似高斯表示的SP-BP相当。我们针对不同数量的智能体和不同通信范围进行了仿真，结果表明，与最先进的参考方法相比，所提出的PF-BP方法具有更优的估计性能。我们还指出了所研究的各类方法在不同场景中的优点与不足。

未来可能的研究方向包括将测量模型扩展到高斯噪声之外，例如考虑漏检、杂波/虚警测量以及测量值的数据关联不确定性 [31], [33], [46], [47], ，或开发基于协作无线电信号、采用高度信息化测量模型的 SLAM 算法 [28], [29], [56]。

附录 A PF 方程的推导

漂移项 $\boldsymbol{\zeta}(\boldsymbol{x}^{(k)},\ \lambda)$ 可利用 FPE 确定，其表达式为

$$\begin{aligned}\frac{\partial f(\boldsymbol{x}^{(k)};\lambda)}{\partial\lambda} =& -\nabla_{\boldsymbol{x}}^{\mathrm{T}}(f(\boldsymbol{x}^{(k)};\lambda)\boldsymbol{\zeta}(\boldsymbol{x}^{(k)},\lambda))\\ &+\frac{1}{2}\nabla_{\boldsymbol{x}}^{\mathrm{T}}(f(\boldsymbol{x}^{(k)};\lambda)\boldsymbol{Q}(\boldsymbol{x}^{(k)},\lambda))\nabla_{\boldsymbol{x}}\end{aligned}\tag{34}$$

其中，$\boldsymbol{Q}(\boldsymbol{x}^{(k)},\ \lambda)$ 对应扩散项。对于 $\boldsymbol{\zeta}(\boldsymbol{x}^{(k)},\ \lambda)$，式（34）的解可分为零扩散情形，即 $\boldsymbol{Q}(\boldsymbol{x}^{(k)},\ \lambda)=0$ [37], [43] 以及非零扩散情形[38], [40]。在进一步推导该方法时，会用到以下两个有用关系：

1）利用散度的链式法则，可将其中的第一项改写为 (34)

$$\begin{aligned}&\nabla_{\boldsymbol{x}}^{\mathrm{T}}(f(\boldsymbol{x}^{(k)};\lambda)\boldsymbol{\zeta}(\boldsymbol{x}^{(k)},\lambda))\\ &=f(\boldsymbol{x}^{(k)};\lambda)\nabla_{\boldsymbol{x}}^{\mathrm{T}}\boldsymbol{\zeta}(\boldsymbol{x}^{(k)},\lambda)+(\nabla_{\boldsymbol{x}}^{\mathrm{T}}f(\boldsymbol{x}^{(k)};\lambda))\boldsymbol{\zeta}(\boldsymbol{x}^{(k)},\lambda).\end{aligned}\tag{35}$$

2）利用式（11），FPE 的左侧，即对 λ的偏导数，可改写为

$$\begin{aligned}&\frac{\partial f(\boldsymbol{x}^{(k)};\lambda)}{\partial\lambda}\\ &=f(\boldsymbol{x}^{(k)}|\boldsymbol{x}^{(k-1)})\left[\frac{\partial f(\boldsymbol{z}^{(k)}|\boldsymbol{x}^{(k)})^{\lambda}}{\partial\lambda}\right]Z(\lambda)^{-1}\\ &\quad+f(\boldsymbol{x}^{(k)}|\boldsymbol{x}^{(k-1)})\ f(\boldsymbol{z}^{(k)}|\boldsymbol{x}^{(k)})^{\lambda}\left[\frac{\partial Z(\lambda)^{-1}}{\partial\lambda}\right]\\ &=f(\boldsymbol{x}^{(k)}|\boldsymbol{x}^{(k-1)})\ f(\boldsymbol{z}^{(k)}|\boldsymbol{x}^{(k)})^{\lambda}\\ &\quad\times\log f(\boldsymbol{z}^{(k)}|\boldsymbol{x}^{(k)})\ Z(\lambda)^{-1}-f(\boldsymbol{x}^{(k)}|\boldsymbol{x}^{(k-1)})\\ &\quad\times f(\boldsymbol{z}^{(k)}|\boldsymbol{x}^{(k)})^{\lambda}Z(\lambda)^{-2}\left[\frac{\partial Z(\lambda)}{\partial\lambda}\right]\\ &=f(\boldsymbol{x}^{(k)};\lambda)\left[\log f(\boldsymbol{z}^{(k)}|\boldsymbol{x}^{(k)})-Z(\lambda)^{-1}\frac{\partial Z(\lambda)}{\partial\lambda}\right]\end{aligned}$$

$$=f(\boldsymbol{x}^{(k)};\lambda)\left[\log f(\boldsymbol{z}^{(k)}|\boldsymbol{x}^{(k)})-\frac{\partial\log Z(\lambda)}{\partial\lambda}\right].\tag{36}$$

假设扩散系数为零，式（34）可简化为

$$\frac{\partial f(\boldsymbol{x}^{(k)};\lambda)}{\partial\lambda}=-\nabla_{\boldsymbol{x}}^{\mathrm{T}}(f(\boldsymbol{x}^{(k)};\lambda)\boldsymbol{\zeta}(\boldsymbol{x}^{(k)},\lambda)).\tag{37}$$

忽略证据 $Z(\lambda)$ 关于λ [37]的导数，并将 (36) 和 (35) 代入 (37)，得到

$$\begin{aligned}\log f(\boldsymbol{z}^{(k)}|\boldsymbol{x}^{(k)}) =& -[f(\boldsymbol{x}^{(k)};\lambda)^{-1}\nabla_{\boldsymbol{x}}^{\mathrm{T}}f(\boldsymbol{x}^{(k)};\lambda)]\boldsymbol{\zeta}(\boldsymbol{x}^{(k)},\lambda)\\ &-\nabla_{\boldsymbol{x}}^{\mathrm{T}}\boldsymbol{\zeta}(\boldsymbol{x}^{(k)},\lambda)\end{aligned}\tag{38}$$

从而得到

$$\begin{aligned}\nabla_{\boldsymbol{x}}^{\mathrm{T}}\boldsymbol{\zeta}(\boldsymbol{x}^{(k)},\lambda) =& -\log f(\boldsymbol{z}^{(k)}|\boldsymbol{x}^{(k)})\\ &-(\nabla_{\boldsymbol{x}}\log f(\boldsymbol{x}^{(k)};\lambda))^{\mathrm{T}}\boldsymbol{\zeta}(\boldsymbol{x}^{(k)},\lambda).\end{aligned}\tag{39}$$

附录B EDH 滤波器的实现

给定 (12) 和 (13)，我们将在此描述 EDH 实现中的状态表示、矩阵和向量。关于 (13)，$\boldsymbol{A}_{\lambda}^{(k)}$ 和 $\boldsymbol{c}_{\lambda}^{(k)}$ 给定为

$$\begin{aligned}\boldsymbol{A}_{\lambda}^{(k)} =& -\frac{1}{2}\boldsymbol{P}^{(k|k-1)}\boldsymbol{H}^{(k)\mathrm{T}}\\ &\times(\lambda\boldsymbol{H}^{(k)}\boldsymbol{P}^{(k|k-1)}\boldsymbol{H}^{(k)\mathrm{T}}+\boldsymbol{R}^{(k)})^{-1}\boldsymbol{H}^{(k)}\end{aligned}\tag{40}$$

$$\begin{aligned}\boldsymbol{c}_{\lambda}^{(k)} =&(\boldsymbol{I}_{N_{\mathrm{D}}|\mathcal{C}|}+2\lambda\boldsymbol{A})\\ &\times[(\boldsymbol{I}_{N_{\mathrm{D}}|\mathcal{C}|}+\lambda\boldsymbol{A})\boldsymbol{P}^{(k|k-1)}\boldsymbol{H}^{(k)\mathrm{T}}(\boldsymbol{R}^{(k)})^{-1}\\ &\times(\boldsymbol{z}^{(k)}+\boldsymbol{\nu}^{(k)})+\boldsymbol{A}\overline{\boldsymbol{x}}_{\lambda=0}^{(k)}]\end{aligned}\tag{41}$$

其中，$\boldsymbol{\nu}^{(k)}=h(\boldsymbol{x}_{\lambda}^{(k)})-\boldsymbol{H}^{(k)}\boldsymbol{x}_{\lambda}^{(k)}$ 和 $\boldsymbol{H}^{(k)}=\frac{\partial h(\boldsymbol{x})}{\partial\boldsymbol{x}}\Big|_{\boldsymbol{x}=\overline{\boldsymbol{x}}_{\lambda}^{(k)}}$，$h(\boldsymbol{x})$ 是一种简写，用于表示所有相连智能体和锚点的所有测量假设；$\boldsymbol{x}_{\lambda}^{(k)}$ 表示伪时间 λ 和时间步 k [43] 下状态的均值。对于 $\lambda=0$，$\boldsymbol{x}_{\lambda=0}^{(k)}$ 对应于提议概率密度函数的均值。根据高斯假设，提议概率密度函数可由预测智能体状态 $\boldsymbol{x}^{(k|k-1)}$ 的均值 $\boldsymbol{x}_{\lambda=0}^{(k)}\triangleq\boldsymbol{x}^{(k|k-1)}$ 和协方差矩阵 $\boldsymbol{P}^{(k|k-1)}$ 完整描述。预测均值和预测协方差矩阵既可以由粒子集合确定，即 $\boldsymbol{x}_{\lambda=0}^{(k)}=(1/M)\sum_{m=1}^{M}\boldsymbol{x}_{\lambda=0}^{(k),m}$和 $\boldsymbol{P}^{(k|k-1)}=(1/M)\sum_{m=1}^{M}(\boldsymbol{x}_{\lambda=0}^{(k),m}-\boldsymbol{x}_{\lambda=0}^{(k)})(\boldsymbol{x}_{\lambda=0}^{(k),m}-\boldsymbol{x}_{\lambda=0}^{(k)})^{\mathrm{T}}$，也可以通过卡尔曼滤波器预测方程确定，具体将在本节后文中介绍。

现在可以确定伪时间步 λ_l、满足 $l\in\{1,\dots,N_{\lambda}\}$ 的联合状态 $\{1/M,\boldsymbol{x}_{\lambda_l}^{(k),m}\}_{m=1}^{M}$ 的粒子表示，其中 N_{λ} 为伪时间步的最大数量；还可以确定该粒子表示的均值：

$$\boldsymbol{x}_{\lambda_l}^{(k),m}=\boldsymbol{x}_{\lambda_{l-1}}^{(k),m}+\boldsymbol{\zeta}(\boldsymbol{x}_{\lambda_{l-1}}^{(k),m},\lambda_l)\Delta_l\tag{42}$$

$$\overline{\boldsymbol{x}}_{\lambda_l}^{(k)}=\overline{\boldsymbol{x}}_{\lambda_{l-1}}^{(k)}+\boldsymbol{\zeta}(\overline{\boldsymbol{x}}_{\lambda_{l-1}}^{(k)},\lambda_l)\Delta_l\tag{43}$$

其中，$\Delta_l=\lambda_l-\lambda_{l-1}$ 为两个连续伪时间步之间流过程的步长。这对应于式 (12) 的解。

为评估流末端（$\lambda=1$）与粒子 (42) 对应的提议分布，我们采用

[43] 中介绍的可逆流原理。根据该原理，根据流末端（$\lambda = 1$）和起始处（$\lambda = 0$）的粒子表示重新计算粒子的权重，即，

$$w^{(k),m} \propto \frac{f(\boldsymbol{x}_{\lambda=1}^{(k),m}|\boldsymbol{x}_{\lambda=0}^{(k),m})\ f(\boldsymbol{z}^{(k)}|\boldsymbol{x}_{\lambda=1}^{(k),m})}{f(\boldsymbol{x}_{\lambda=0}^{(k),m})}. \tag{44}$$

这里，$\boldsymbol{x}^{(k)}{}_{,\lambda=0}^{\ m}$ 是从提议概率密度函数中采样得到的粒子，该函数由高斯分布表示。联合智能体状态$\mathbf{x}^{(k)}$ 的后验概率密度函数随后由加权粒子集合 $\{w^{(k),m}, \boldsymbol{x}_{\lambda=1}^{(k),m}\}_{m=1}^{M}$表示。最后，我们对联合状态进行系统重采样，得到时刻 k 联合智能体状态的后验概率密度函数 $\{1/M, \boldsymbol{x}^{(k),m}\}_{m=1}^{M}$ [36]，其中省略了索引 λ。

与 [43] 类似，我们基于粒子表示的后验概率密度函数 $\boldsymbol{x}_{\lambda=1}^{(k)} = (1/M)\ \sum_{m=1}^{M} \boldsymbol{x}^{(k)}{}_{,\lambda=1}^{\ m}$ 的基于样本的均值处的无迹卡尔曼滤波器（UKF）更新步骤[25], [57]，计算后验协方差矩阵$\boldsymbol{P}^{(k)}$；同时使用预测协方差 $\boldsymbol{P}_{(k|k-1)}$。预测协方差矩阵为

$$\boldsymbol{P}^{(k|k-1)} = \tilde{\boldsymbol{F}}\boldsymbol{P}^{(k-1)}\tilde{\boldsymbol{F}}^{\mathrm{T}} + \boldsymbol{W} \tag{45}$$

其中

$$\tilde{\boldsymbol{F}} = \boldsymbol{I}_{|\mathcal{C}|} \otimes \boldsymbol{F} \tag{46}$$

$$\boldsymbol{W} = \boldsymbol{I}_{|\mathcal{C}|} \otimes \boldsymbol{Q} \tag{47}$$

$$\boldsymbol{Q} = \boldsymbol{G}(\boldsymbol{I}_3 \odot \sigma_a^2)\boldsymbol{G}^{\mathrm{T}} \tag{48}$$

更新步骤如下

$$\boldsymbol{P}^{(k)} = \boldsymbol{P}^{(k|k-1)} - \boldsymbol{K}\boldsymbol{P}_{zz}\boldsymbol{K}^{\mathrm{T}} \tag{49}$$

其中，$\boldsymbol{K}$ 为 [25], [57]中定义的卡尔曼增益，$\boldsymbol{P}_{zz}$为测量协方差矩阵。有关无迹卡尔曼滤波器的更多详细信息，见 [25], [57]。

通过比较 $\boldsymbol{R}^{(k)}$ 和 $\boldsymbol{P}^{(k|k-1)}$的秩，可以缩短 EDH 滤波器的计算时间。如果 $\boldsymbol{R}^{(k)}$ 的秩大于 $\boldsymbol{P}^{(k|k-1)}$的秩，则利用伍德伯里矩阵恒等式改写式（40）。

参考文献

[1] L. D. Xu、W. He 和 S. Li，“工业中的物联网：综述”，*IEEE Trans. Ind. Informat.*，第 10 卷，第 4 期，第 2233–2243 页，2014 年。[2] K. Witrisal、P. Meissner、E. Leitinger、Y. Shen、C. Gustafson、F. Tufvesson、K. Haneda、D. Dardari、A. F. Molisch、A. Conti 和 M. Z. Win，“辅助生活的高精度定位：5G系统将使多径信道化敌为友”，*IEEE Signal Process. Mag.*，第 33 卷，第 2 期，第 59–70 页，2016 年 3 月。[3] M. Z. Win、F. Meyer、Z. Liu、W. Dai、S. Bartoletti 和 A. Conti，“物联网的高效多传感器定位：探索一类新型可扩展定位算法”，*IEEE Signal Process. Mag.*，第 35 卷，第 5 期，第 153–167 页，2018 年 9 月。[4] R. Di Taranto、S. Muppirisetty、R. Raulefs、D. Slock、T. Svensson 和 H. Wymeersch，“面向5G网络的位置感知通信：位置信息如何提升5G的可扩展性、时延和鲁棒性”，*IEEESignalProcess.Mag.*，第 31 卷，第 6 期，第 102–112 页，2014 年 11 月。[5] N. Patwari、J. N. Ash、S. Kyperountas、A. O. Hero、R. L. Moses 和 N. S. Correal，“定位节点：无线传感器网络中的协作定位”，*IEEE Signal Process. Mag.*，第 22 卷，第 4 期，第 54–69 页，2005 年 7 月。[6] Y. Shen、H. Wymeersch 和 M. Win，“宽带定位的基本极限；第二部分：协作网络”，*IEEE Trans. Inf. Theory*，第 56 卷，第 10 期，第 4981–5000 页，2010 年 10 月。[7] Y. Shen、S. Mazuelas 和 M. Win，“网络导航：理论与阐释”，*IEEE J. Sel. Areas Commun.*，第 30 卷，第 9 期，第 1823–1834 页，2012 年 10 月。

[8] M. Z. Win、Y. Shen 和 W. Dai，《网络定位与导航的理论基础》，《IEEE会刊》，第106卷，第7期，第1136–1165页，2018年7月。[9] J. Kulmer、E. Leitinger、S. Grebien 和 K. Witrisal，《利用多径信道信息进行无锚点协作跟踪》，《IEEE无线通信汇刊》，第17卷，第4期，第2262–2275页，2018年4月。[10] H. Ping，《IMF：一种用于可定位性检测的迭代最大流方法》，CoRR，第abs/2102.07100卷，2021年。 [在线]。可访问：https://arxiv.org/abs/2102.07100 N. Alsindi 和 K. Pahlavan，《室内超宽带无线传感器网络的协作定位界》，《EURASIP信号处理进展期刊》，第2008卷，第1期，第852509页，2007年12月。X. Mei、H. Wu 和 J. Xian，《在发射功率不确定条件下通过距离测量实现的基于矩阵分解的目标定位》，《IEEE无线通信快报》，第9卷，第10期，第1611–1615页，2020年。X. Mei、H. Wu、J. Xian 和 B. Chen，《非视距环境下基于RSS的拜占庭容错定位算法》，《IEEE通信快报》，第25卷，第2期，第474–478页，2021年。H. Wu、L. Liang、X. Mei 和 Y. Zhang，《基于到达时间的定位中非视距误差抑制的凸优化方法》，《IEEE信号处理快报》，第29卷，第677–681页，2022年。[15] Y. Zhang、H. Wu、X. Mei、L. Liang 和 T. A. Gulliver，《高斯混合信道中发射功率未知的基于RSSD的定位》，《IEEE传感器期刊》，第22卷，第9期，第9114–9123页，2022年。[16] L. Wielandner、E. Leitinger 和 K. Witrisal，《利用天线方向性进行基于RSS的协作定位与方向估计》，《IEEE Access》，第9卷，第53 046–53 060页，2021年3月。[17] ——，《一种利用置信传播进行联合协作定位与方向估计的自适应算法》，载于《2021年第55届阿西洛马会议：信号、系统与计算机》，2021年11月，第1591–1596页。[18] H. Wymeersch、J. Lien 和 M. Z. Win，《无线网络中的协作定位》，《IEEE会刊》，第97卷，第2期，第427–450页，2009年2月。[19] D. Dardari、P. Closas 和 P. M. Djuri´c，《室内跟踪：理论、方法与技术》，《IEEE车辆技术汇刊》，第64卷，第4期，2015年。[20] F. Meyer、O. Hlinka、H. Wymeersch、E. Riegler 和 F. Hlawatsch，《包含非协作对象的移动网络分布式定位与跟踪》，《IEEE信号信息处理网络汇刊》，第2卷，第1期，第57–71页，2016年3月。[21] B. Cakmak、D. N. Urup、F. Meyer、T. Pedersen、B. H. Fleury 和 F. Hlawatsch，《移动网络协作定位：一种分布式置信传播–平均场消息传递算法》，《IEEE信号处理快报》，第23卷，第6期，第828–832页，2016年4月。[22] C. Pedersen、T. Pedersen 和 B. H. Fleury，《一种用于无线网络传感器自定位的变分消息传递算法》，载于《2011年IEEE国际信息论研讨会论文集》，2011年，第2158–2162页。[23] A. T. Ihler、J. W. Fisher、R. L. Moses 和 A. S. Willsky，《用于传感器网络自定位的非参数置信传播》，《IEEE精选通信领域期刊》，第23卷，第4期，第809–819页，2005年4月。[24] J. Lien、U. J. Ferner、W. Srichavengsup、H. Wymeersch 和 M. Z. Win，《协作定位中参数化与基于样本的消息表示比较》，《国际导航与观测期刊》，第2012卷，2012年。[25] F. Meyer、O. Hlinka 和 F. Hlawatsch，《西格玛点置信传播》，《IEEE信号处理快报》，第21卷，第2期，第145–149页，2013年2月。[26] M. Liang 和 F. Meyer，《用于协作定位的神经网络增强置信传播》，载于《2021年IEEE统计信号处理研讨会（SSP）》，2021年，第326–330页。[27] E. Leitinger、F. Meyer、F. Hlawatsch、K. Witrisal、F. Tufvesson 和 M. Z. Win，《一种用于基于多径的同步定位与建图的置信传播算法》，《IEEE无线通信汇刊》，第18卷，第12期，第5613–5629页，2019年9月。[28] E. Leitinger、S. Grebien 和 K. Witrisal，《利用到达角和幅度信息的基于多径的同步定位与建图》，载于《IEEE ICCW-19论文集》，中国上海，2019年5月，第1–7页。[29] E. Leitinger 和 F. Meyer，《基于多径的同步定位与建图的数据融合》，载于《阿西洛马-20论文集》，美国加利福尼亚州太平洋丛林，2020年10月，第934–939页。[30] F. Meyer、T. Kropfreiter、J. L. Williams、R. Lau、F. Hlawatsch、P. Braca 和 M. Z. Win，《可扩展多目标跟踪的消息传递算法》，《IEEE会刊》，第106卷，第2期，第221–259页，2018年2月。[31] D. Gaglione、P. Braca、G. Soldi、F. Meyer、F. Hlawatsch 和 M. Z. Win，《多目标跟踪中传感器测量与目标提供信息的融合》，《IEEE信号处理汇刊》，第70卷，第322–336页，2022年12月。[32] F. Meyer 和 J. L. Williams，《几何扩展目标的可扩展检测与跟踪》，《IEEE信号处理汇刊》，第69卷，第6283–6298页，2021年10月。[33] M. Brambilla、D. Gaglione、G. Soldi、R. Mendrzik、G. Ferri、K. D. LePage、M. Nicoli、P. Willett、P. Braca 和 M. Z. Win，《智能体网络中采用乘积算法的协作定位与多目标跟踪，

参考文献

[1] 《IEEE信号处理期刊》，第3卷，第169–195页，2022年3月。[34] T. Bengtsson、P. Bickel和B. Li，“重新审视维数灾难：超大规模系统中的粒子滤波器退化”，载于《概率与统计：纪念David A. Freedman文集》。数理统计学会，2008年，第316–334页。[35] C. Musso、N. Oudjane和F. Le Gland，“改进正则化粒子滤波器”，载于《实践中的序贯蒙特卡洛方法》。施普林格，纽约，纽约州，2001年，第247–271页。[36] M. Arulampalam、S. Maskell、N. Gordon和T. Clapp，“在线非线性/非高斯贝叶斯跟踪的粒子滤波器教程”，《IEEE信号处理汇刊》，第50卷，第2期，第174–188页，2002年2月。[37] F. Daum、J. Huang和A. Noushin，“非线性滤波器的精确粒子流”，载于《国际光学工程学会会议录-10》，第7697卷，2010年4月，第769704页。[38] F. Daum和J. Huang，“具有非零扩散的非线性滤波器粒子流”，载于《国际光学工程学会会议录-13》，第8745卷，2013年5月，第87450P页。[39] F. Daum、J. Huang和A. Noushin，“随机粒子流滤波器的广义格罗莫夫方法”，载于《国际光学工程学会会议录-17》，第10200卷，2017年5月，第102000I页。[40] ——，“随机粒子流滤波器格罗莫夫方法的新理论与数值结果”，载于《IEEE Fusion-18会议录》，英国剑桥，2018年7月，第108–115页。[41] Y. Li、S. Pal和M. J. Coates，“基于可逆粒子流的序贯马尔可夫链蒙特卡洛及其在高斯混合噪声模型中的扩展”，《IEEE信号处理汇刊》，第67卷，第9期，第2499–2512页，2019年。[42] S. Pal、L. Ma、Y. Zhang和M. Coates，“用于概率时空预测的结合粒子流的循环神经网络”，载于《第38届国际机器学习会议论文集》，机器学习研究论文集，M. Meila和T. Zhang编，第139卷。机器学习研究论文集，2021年7月18–24日，第8336–8348页。[43] Y. Li和M. Coates，“利用可逆粒子流进行粒子滤波”，《IEEE信号处理汇刊》，第65卷，第15期，第4102–4116页，2017年5月。[44] W. Zhang和F. Meyer，“嵌入粒子流的基于图的多目标跟踪”，载于《IEEE RadarConf-21会议录》，美国佐治亚州亚特兰大，2021年，第1–6页。[45] P. Tichavsky、C. Muravchik和A. Nehorai，“离散时间非线性滤波的后验克拉美-罗界”，《IEEE信号处理汇刊》，第46卷，第5期，第1386–1396页，1998年5月。[46] A. Venus、E. Leitinger、S. Tertinek和K. Witrisal，“基于消息传递的自适应概率数据关联算法，用于稳健的基于无线电的定位与跟踪”，载于《2021年IEEE RadarConf-21会议录》，2021年6月，第1–6页。[47] F. Meyer和M. Z. Win，“用于扩展目标跟踪的可扩展数据关联”，《IEEE信号与信息处理网络汇刊》，第6卷，第491–507页，2020年5月。[48] L. Wielandner、E. Leitinger、F. Meyer、B. Teague和K. Witrisal，“嵌入粒子流的基于消息传递的协作定位”，载于《ICASSP 2022——2022年IEEE声学、语音与信号处理国际会议（ICASSP）》，2022年5月，第5652–5656页。[49] S. M. Kay，《统计信号处理基础：估计理论》。美国新泽西州上萨德尔河：普伦蒂斯·霍尔，1993年。[50] F. Kschischang、B. Frey和H.-A. Loeliger，“因子图与和积算法”，《IEEE信息论汇刊》，第47卷，第2期，第498–519页，2001年2月。[51] H.-A. Loeliger，“因子图导论”，《IEEE信号处理杂志》，第1期，第28–41页，2004年1月。[52] A. Doucet、N. de Freitas和N. Gordon，《实践中的序贯蒙特卡洛方法》。施普林格，2001年。[53] D. F. Crouse和C. Lewis，“对粒子流滤波器实现及其偏差的考察”，美国华盛顿特区海军研究实验室，技术报告，2020年。[54] H. Risken，“福克-普朗克方程”，载于《福克-普朗克方程》。施普林格，1996年，第63–95页。[55] F. Meyer、H. Wymeersch、M. Frohle和F. Hlawatsch，“智能体网络中结合信息寻求控制的分布式估计”，《IEEE通信领域精选期刊》，第33卷，第11期，第2439–2456页，2015年11月。[56] H. Kim、K. Granstr¨om、L. Gao、G. Battistelli、S. Kim和H. Wymeer- sch，“利用多模型PHD滤波器和地图融合实现5G毫米波协作定位与制图”，《IEEE无线通信汇刊》，第19卷，第6期，第3782–3795页，2020年3月。[57] E. Wan和R. Van Der Merwe，“用于非线性估计的无迹卡尔曼滤波器”，载于《IEEE 2000年信号处理、通信与控制自适应系统研讨会论文集》（目录号00EX373），2000年10月，第153–158页。