A Distributed Particle-PHD Filter Using Arithmetic-Average Fusion of Gaussian Mixture Parameters

Tiancheng Li[a], Franz Hlawatsch[b]

[a]*Key Laboratory of Information Fusion Technology (Ministry of Education), School of Automation, Northwestern Polytechnical University, Xi'an 710072, China. E-mail: t.c.li@nwpu.edu.cn*

[b]*Institute of Telecommunications, TU Wien, 1040 Vienna, Austria. E-mail: franz.hlawatsch@tuwien.ac.at*

Abstract

We propose a particle-based distributed PHD filter for tracking the states of an unknown, time-varying number of targets. To reduce communication, the local PHD filters at neighboring sensors communicate Gaussian mixture (GM) parameters. In contrast to most existing distributed PHD filters, our filter employs an "arithmetic average" fusion. For particles–GM conversion, we use a method that avoids particle clustering and enables a significance-based pruning of the GM components. For GM–particles conversion, we develop an importance sampling based method that enables a parallelization of filtering and dissemination/fusion operations. The resulting distributed PHD filtering framework is able to integrate both particle-based and GM-based local PHD filters. Simulations demonstrate the excellent performance and small communication and computation requirements of our filter.

Keywords: Distributed multitarget tracking, distributed PHD filter, arithmetic average fusion, average consensus, flooding, probability hypothesis density, random finite set, Gaussian mixture, sequential Monte Carlo, importance sampling.

1. Introduction

1.1. Background and State-of-the-Art

The probability hypothesis density (PHD) filter is a popular method for tracking the states of an unknown, time-varying number of targets in the presence of clutter and missed detections [1, 2, 3]. In decentralized sensor networks, a distributed extension of the PHD filter can be employed where each sensor runs a local PHD filter and performs *distributed sensor fusion* by exchanging relevant information with neighboring sensors. For the local PHD filters, a Gaussian mixture (GM) implementation [4, 5, 6, 7, 8, 9, 10] or a particle-based implementation [11, 12, 13, 14] is typically used. A particle-based implementation is advantageous because of its maximal flexibility and suitability for arbitrary nonlinear and/or non-Gaussian system models.

For distributed sensor fusion, most existing distributed PHD filters perform a "geometric average" (GA) fusion of the local posterior PHDs [11, 12, 4, 5, 6, 7]; this type of fusion is also known as (generalized) covariance intersection [15, 16, 17, 18, 19, 20] or log-linear fusion [21]. GA fusion minimizes the sum of the Kullback-Leibler divergences of the result of fusion relative to the local posterior PHDs [21]. Furthermore, it does not double-count information and is thus able to deal with unknown correlation between sensors, which is sometimes referred to as "conservative" fusion [17]. On the other hand, GA fusion of PHDs has been observed to suffer from certain deficiencies in practical implementations: it performs poorly in the case of closely spaced targets [9, 10]; incurs a delay in detecting new targets [6, 10]; is sensitive to missing measurements or missed detections [8, 7, 22]; and tends to underestimate the number of targets [23, 24].

An alternative to GA fusion is the linear fusion given by the arithmetic average (AA) of the local posterior PHDs. AA fusion of PHDs first appeared indirectly in the context of centralized multisensor PHD filtering, as an implicit consequence of AA fusion of the generalized likelihood functions of multiple sensors [25]. It was used explicitly for distributed PHD filtering in [8, 10, 22, 26] (based on a GM implementation of the local PHD filters) and in [13, 14] (based on a particle implementation of the local PHD filters and a straightforward particle-based dissemination/fusion scheme). Similar to GA fusion, AA fusion minimizes the sum of the Kullback-Leibler divergences of the local posterior PHDs relative to the result of fusion [21, 27], and it is also conservative. The results of both AA and GA fusion can be interpreted as a Fréchet mean [28]; this interpretation as well as the conservativeness properties of AA and GA fusion are discussed in [29].

AA fusion tends to cause an increased spread (variance) of the fusion result [30, 31, 17, 32]. This may lead to a reduced estimation accuracy in certain cases, but tends to increase the robustness of estimation. More specifically, AA fusion is able to compensate for imperfections of a local posterior PHD such as an offset relative to a true target position, a missed detection (i.e., a target is not accounted for by a local posterior PHD), or a spurious PHD component (suggesting the presence of a target that is actually not present) by using the information provided by the other local posterior PHDs. AA fusion of PHDs was demonstrated in [8, 10, 13, 14] (and in certain cases in [22]) to outperform GA fusion of PHDs in the sense of better filtering accuracy, higher reliability in scenarios with frequent missed detections, and/or lower computational complexity. Advantages of AA fusion over GA fusion have also been observed in the context of PHD-based joint sensor localization and target tracking [33] and in the context of particle filtering [34]. An extensive case-study comparison of AA and GA fusion of PHDs is provided in [32], and further empirical arguments in favor of AA fusion can be found in [29].

1.2. Contribution and Paper Organization

In this paper, we propose a distributed PHD filter method that performs AA fusion of the local posterior PHDs. The local PHD filters employ particle implementations for the sake of maximum suitability for nonlinear and/or non-Gaussian system models. Although the particle-PHD filter is a popular multitarget tracking method, its distributed implementation is challenging because straightforward fusion of particle representations of the local PHDs imposes high communication requirements [35, 12, 13, 14]. By contrast, our distributed particle-PHD filter has low communication requirements because GM parameters are communicated. This also allows our particle-based local PHD filters to be easily combined with GM-based local PHD filters within a heterogeneous network architecture.

The overall contribution of this paper is to devise an AA fusion-based distributed particle-PHD filter that has *low communication requirements* and allows for a *parallelization of filtering and dissemination/fusion operations*. Specific contributions and innovative aspects of our paper that constitute an advance over the current state of the art—including our previous work [10, 14]—are as follows:

- We present the first distributed particle-PHD filter using AA fusion that is able to achieve a network-wide consensus on the fused PHD. This filter differs from the distributed GM-PHD filters proposed in [11, 12, 4, 5, 6, 7] (which use GA fusion) and proposed in [10] (which use AA fusion). It also differs from the distributed particle-PHD filter proposed in [14], which performs AA fusion by communicating particles merely between neighbor sensors and is unable to achieve a networkwide consensus.

- For converting particle representations into GM representations, we propose a data-driven method that avoids a clustering of the particles. This method generates from the particle representation one Gaussian component for each measurement that has a significant influence on the particle weights. The overall approach is inspired by a scheme for estimate extraction that was proposed in [36, 37, 38] and was demonstrated there to be more accurate and also less complex than methods based on particle clustering [2].
- For converting the GMs produced by AA fusion into particle representations, we propose a method that is based on importance sampling (IS) [39, Ch. 3.3]. This method does not require sampling from the fused GM, thereby enabling a parallelization of filtering and dissemination/fusion operations. This allows more dissemination/fusion iterations to be performed compared to protocols where the filtering and dissemination/fusion operations must be performed serially. The proposed parallelization approach can also be employed for general distributed particle filters.
- We propose the use of mixture merging in iterative AA fusion of PHDs for distributed particle-PHD filtering. Mixture merging reduces the "spread enhancement" of AA fusion and leads to an improved performance of distributed PHD filters using AA fusion.
- We demonstrate experimentally that the proposed distributed particle-PHD filter using AA fusion with mixture merging outperforms the state-of-the-art GA fusion approach and at the same time has much lower computation requirements.

The paper is organized as follows. The system model is described in Section 2. Section 3 discusses the basic operation of the particle-based local PHD filters and presents a measurement-based weight decomposition. Section 4 provides a motivation and outline of the proposed distributed PHD filter. Section 5 describes a method for converting particle representations into GM representations. Section 6 discusses two schemes for GM dissemination and fusion. An IS method for converting the fused GM into a particle representation is proposed in Section 7. Section 8 presents two further stages of the proposed distributed PHD filter. Section 9 provides a summary of the overall method, discusses the parallelization of filtering and fusion, and analyzes the communication cost. Simulation results are reported in Section 10.

2. System Model

We consider N_k targets with random states $\mathbf{x}_k^{(\nu)} \in \mathbb{R}^d$, $\nu = 1, 2, \ldots, N_k$ at discrete time k. The number of targets, N_k, is unknown, time-varying, and considered random. Accordingly, the collection of target states is modeled by a random finite set (RFS) $X_k = \{\mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)}, \ldots, \mathbf{x}_k^{(N_k)}\}$ with random cardinality $N_k = |X_k|$ [40]. A target with state $\mathbf{x}_{k-1}$ at time $k-1$ continues to exist at time k with probability $p_k^{\mathrm{S}}(\mathbf{x}_{k-1})$ ("survival probability") or disappears with probability $1 - p_k^{\mathrm{S}}(\mathbf{x}_{k-1})$. In the first case, its new state $\mathbf{x}_k \in X_k$ is distributed according to a transition probability density function (pdf) $f_k(\mathbf{x}_k|\mathbf{x}_{k-1})$. There may also be newborn targets, whose states are modeled by a Poisson RFS with intensity function $\gamma_k(\mathbf{x}_k)$ [41].

There are S sensors indexed by $s \in \{1, 2, \ldots, S\}$. At time k, each sensor s observes an RFS of measurements $Z_{s,k} = \{\mathbf{z}_{s,k}^{(1)}, \ldots, \mathbf{z}_{s,k}^{(M_{s,k})}\}$, where $M_{s,k}$ is the number of measurements observed by sensor s at time k. We denote by $\mathcal{S}_s \subseteq \{1, 2, \ldots, S\} \setminus \{s\}$ the set of sensors that are connected to sensor s by a communication

link, and we refer to these sensors as the neighbors of sensor s. We assume that the sensor network is connected, i.e., each sensor can be reached from each other sensor by one or multiple communication hops. A target with state $\mathbf{x}_k$ is "detected" by sensor s with probability $p^{\mathrm{D}}_{s,k}(\mathbf{x}_k)$ ("detection probability") or "missed" by sensor s with probability $1-p^{\mathrm{D}}_{s,k}(\mathbf{x}_k)$. In the first case, the target generates a measurement $\mathbf{z}_k \in Z_k$, which is distributed according to the pdf $g_{s,k}(\mathbf{z}_k|\mathbf{x}_k)$. There may also be clutter measurements, which are modeled by a Poisson RFS with intensity function (PHD) $\kappa_{s,k}(\mathbf{z}_k)$. The multitarget state evolution and measurement processes are assumed to satisfy the independence assumptions discussed in [1].

3. Local Particle-PHD Filters

Each sensor runs a local PHD filter that uses the local measurement set $Z_{s,k}$ and communicates with its neighbors $r \in \mathcal{S}_s$ to exchange relevant information. Let us, at first, consider the local PHD filter without any cooperation.

3.1. Propagation of the Local Posterior PHD

The local PHD filter propagates the local posterior PHD over time k. Let $Z_{s,1:k} = (Z_{s,1}, \ldots, Z_{s,k})$ comprise the local measurements $Z_{s,k'}$ observed by sensor s up to time k. Furthermore, for a region $\mathcal{R} \subseteq \mathbb{R}^d$, let $N^{\mathcal{R}}_k \triangleq |X_k \cap \mathcal{R}|$ denote the number of those targets whose states are in $\mathcal{R}$. Then, the local posterior PHD at sensor s, $D_{s,k}(\mathbf{x}|Z_{s,1:k})$, is defined as the function of $\mathbf{x} \in \mathbb{R}^d$ whose integral over a region $\mathcal{R} \subseteq \mathbb{R}^d$ equals the posterior expectation of $N^{\mathcal{R}}_k$, i.e. [40]

$$\int_{\mathcal{R}} D_{s,k}(\mathbf{x}|Z_{s,1:k})\,\mathrm{d}\mathbf{x} = \mathrm{E}[N^{\mathcal{R}}_k|Z_{s,1:k}]. \tag{1}$$

In particular, for $\mathcal{R} = \mathbb{R}^d$, we have $N^{\mathbb{R}^d}_k = |X_k \cap \mathbb{R}^d| = |X_k| = N_k$, and thus (1) becomes

$$\int_{\mathbb{R}^d} D_{s,k}(\mathbf{x}|Z_{s,1:k})\,\mathrm{d}\mathbf{x} = \mathrm{E}[N_k|Z_{s,1:k}] = \sum_{n=0}^{\infty} n\,\rho(n|Z_{s,1:k}), \tag{2}$$

where $\rho(n|Z_{s,1:k}) = \Pr[N_k = n|Z_{s,1:k}]$. The posterior expectation of N_k, $\mathrm{E}[N_k|Z_{s,1:k}]$, is equal to the minimum mean square error (MMSE) estimate of N_k from $Z_{s,1:k}$ [42], denoted $\hat{N}^{\mathrm{MMSE}}_{s,k}$. Thus, Eq. (2) implies

$$\hat{N}^{\mathrm{MMSE}}_{s,k} = \mathrm{E}[N_k|Z_{s,1:k}] = \int_{\mathbb{R}^d} D_{s,k}(\mathbf{x}|Z_{s,1:k})\,\mathrm{d}\mathbf{x}. \tag{3}$$

This is also known as the expected a posteriori (EAP) estimate of N_k [1, 40].

The local PHD filter performs a time-recursive calculation of an approximation $\hat{D}_{s,k}(\mathbf{x}|Z_{s,1:k})$ of the local posterior PHD $D_{s,k}(\mathbf{x}|Z_{s,1:k})$. In a *prediction step*, it converts the preceding approximate local posterior PHD $\hat{D}_{s,k-1}(\mathbf{x}|Z_{s,1:k-1})$ into a "predicted" PHD, denoted $D_{s,k|k-1}(\mathbf{x}|Z_{s,1:k-1})$, via an expression involving $f_k(\mathbf{x}_k|\mathbf{x}_{k-1})$, $p^{\mathrm{S}}_k(\mathbf{x})$, and $\gamma_k(\mathbf{x})$ [1]. In a subsequent *update step*, it converts $D_{s,k|k-1}(\mathbf{x}|Z_{s,1:k-1})$ into the approximate local posterior PHD $\hat{D}_{s,k}(\mathbf{x}|Z_{s,1:k})$ via an expression involving $g_{s,k}(\mathbf{z}|\mathbf{x})$, $p^{\mathrm{D}}_{s,k}(\mathbf{x})$, and $\kappa_{s,k}(\mathbf{z})$ [1, 2]. In a stand-alone PHD filter, the approximate local posterior PHD $\hat{D}_{s,k}(\mathbf{x}|Z_{s,1:k})$ would finally be used for estimating the number and states of the targets. However, in a distributed PHD filter, this estimation is done only after networkwide fusion of the local posterior PHDs.

3.2. Particle-Based Implementation

We use the particle-based implementation of the prediction and update steps proposed in [2]. The approximate local posterior PHD $\hat{D}_{s,k}(\mathbf{x}|Z_{s,1:k})$ is represented by the weighted particle set $\xi_{s,k} \triangleq \{(\mathbf{x}_{s,k}^{(j)}, w_{s,k}^{(j)})\}_{j=1}^{J_{s,k}}$, which consists of $J_{s,k}$ particles $\mathbf{x}_{s,k}^{(j)} \in \mathbb{R}^d$ and weights $w_{s,k}^{(j)} \geq 0$, $j = 1, 2, \ldots, J_{s,k}$. The sum of the weights,

$$W_{s,k} \triangleq \sum_{j=1}^{J_{s,k}} w_{s,k}^{(j)}, \tag{4}$$

approximates $\int_{\mathbb{R}^d} \hat{D}_{s,k}(\mathbf{x}|Z_{s,1:k})\mathrm{d}\mathbf{x}$ and, hence, $\int_{\mathbb{R}^d} D_{s,k}(\mathbf{x}|Z_{s,1:k})\mathrm{d}\mathbf{x}$. Thus, it further follows with (3) that

$$W_{s,k} \approx \hat{N}_{s,k}^{\mathrm{MMSE}}. \tag{5}$$

Propagating the approximate local posterior PHD (i.e., $\hat{D}_{s,k-1}(\mathbf{x}|Z_{s,1:k-1}) \rightarrow \hat{D}_{s,k}(\mathbf{x}|Z_{s,1:k})$) is now approximated by propagating the weighted particle set, i.e., $\xi_{s,k-1} \rightarrow \xi_{s,k}$. The time-recursive calculation of $\xi_{s,k}$ is done as follows [2]. For each previous particle $\mathbf{x}_{s,k-1}^{(j)}$, $j \in \{1, \ldots, J_{s,k-1}\}$, a current particle $\mathbf{x}_{s,k}^{(j)}$ is drawn from a proposal pdf $q_{s,k}(\mathbf{x}; \mathbf{x}_{s,k-1}^{(j)}, Z_{s,k})$. In addition, $L_{s,k} \triangleq J_{s,k} - J_{s,k-1}$ "newborn" particles $\mathbf{x}_{s,k}^{(j)}$, $j = J_{s,k-1} + 1, \ldots, J_{s,k}$ are drawn from a proposal pdf $p_{s,k}(\mathbf{x}; Z_{s,k})$. Then, for each particle $\mathbf{x}_{s,k}^{(j)}$, $j \in \{1, \ldots, J_{s,k}\}$, a "predicted" weight $w_{s,k|k-1}^{(j)}$ is calculated as

$$w_{s,k|k-1}^{(j)} = \begin{cases} \dfrac{f_k(\mathbf{x}_{s,k}^{(j)}|\mathbf{x}_{s,k-1}^{(j)}) w_{s,k-1}^{(j)}}{q_{s,k}(\mathbf{x}_{s,k}^{(j)}; \mathbf{x}_{s,k-1}^{(j)}, Z_{s,k})}, & j = 1, \ldots, J_{s,k-1}, \\ \dfrac{\gamma_k(\mathbf{x}_{s,k}^{(j)})}{L_{s,k}\, p_{s,k}(\mathbf{x}_{s,k}^{(j)}; Z_{s,k})}, & j = J_{s,k-1}+1, \ldots, J_{s,k}. \end{cases} \tag{6}$$

Note that $J_{s,k} = J_{s,k-1} + L_{s,k}$. A simple choice of the first proposal pdf is $q_{s,k}(\mathbf{x}; \mathbf{x}_{s,k-1}^{(j)}, Z_{s,k}) = f_k(\mathbf{x}|\mathbf{x}_{s,k-1}^{(j)})$, in which case $w_{s,k|k-1}^{(j)} = w_{s,k-1}^{(j)}$ for $j = 1, \ldots, J_{s,k-1}$.

For the calculation of the current weights $w_{s,k}^{(j)}$, $j = 1, \ldots, J_{s,k}$, we formally introduce a dummy vector $\mathbf{z}_0$ representing the case of a missed detection at sensor s, and, accordingly, we consider an *extended measurement set* $Z_{s,k}^0 \triangleq \{\mathbf{z}_0\} \cup Z_{s,k} = \{\mathbf{z}_0, \mathbf{z}_{s,k}^{(1)}, \ldots, \mathbf{z}_{s,k}^{(M_{s,k})}\}$. Then, the weight expression in [2, Eq. (22)] can be formulated as the sum [36, 37, 38]

$$w_{s,k}^{(j)} = \sum_{\mathbf{z} \in Z_{s,k}^0} \omega_{s,k}^{(j)}(\mathbf{z}), \quad j = 1, \ldots, J_{s,k}, \tag{7}$$

where

$$\omega_{s,k}^{(j)}(\mathbf{z}) = \begin{cases} (1 - p_{s,k}^{\mathrm{D}}(\mathbf{x}_{s,k}^{(j)})) w_{s,k|k-1}^{(j)}, & \mathbf{z} = \mathbf{z}_0 \\ \dfrac{p_{s,k}^{\mathrm{D}}(\mathbf{x}_{s,k}^{(j)}) g_{s,k}(\mathbf{z}|\mathbf{x}_{s,k}^{(j)}) w_{s,k|k-1}^{(j)}}{\kappa_{s,k}(\mathbf{z}) + G_{s,k}(\mathbf{z})}, & \mathbf{z} \in Z_{s,k}, \end{cases} \tag{8}$$

with $G_{s,k}(\mathbf{z}) \triangleq \sum_{j=1}^{J_{s,k}} p_{s,k}^{\mathrm{D}}(\mathbf{x}_{s,k}^{(j)}) g_{s,k}(\mathbf{z}|\mathbf{x}_{s,k}^{(j)}) w_{s,k|k-1}^{(j)}$. Expression (7) provides an expansion of $w_{s,k}^{(j)}$ into $|Z_{s,k}^0| = M_{s,k} + 1$ components $\omega_{s,k}^{(j)}(\mathbf{z})$, each of which corresponds to one of the measurements $\mathbf{z} \in Z_{s,k}^0$. We also introduce

$$\Omega_{s,k}(\mathbf{z}) \triangleq \sum_{j=1}^{J_{s,k}} \omega_{s,k}^{(j)}(\mathbf{z}), \quad \mathbf{z} \in Z_{s,k}^0. \tag{9}$$

For $\mathbf{z} \in Z_{s,k}$, $\Omega_{s,k}(\mathbf{z}) = G_{s,k}(\mathbf{z})/(\kappa_{s,k}(\mathbf{z}) + G_{s,k}(\mathbf{z})) \in [0,1]$, which provides an estimate of the probability that measurement $\mathbf{z}$ originates from a target. For $\mathbf{z} = \mathbf{z}_0$, $\Omega_{s,k}(\mathbf{z}_0) = \sum_{j=1}^{J_{s,k}} (1 - p_{s,k}^{\mathrm{D}}(\mathbf{x}_{s,k}^{(j)}))w_{s,k|k-1}^{(j)}$ provides an estimate of the number of missed detections. Note that

$$\sum_{\mathbf{z} \in Z_{s,k}^0} \Omega_{s,k}(\mathbf{z}) = \sum_{j=1}^{J_{s,k}} \sum_{\mathbf{z} \in Z_{s,k}^0} \omega_{s,k}^{(j)}(\mathbf{z}) = \sum_{j=1}^{J_{s,k}} w_{s,k}^{(j)} = W_{s,k}, \tag{10}$$

where (7) and (4) were used.

4. Motivation and Outline of the Proposed PHD Fusion Scheme

The proposed distributed PHD filter uses information fused among the sensors to "re-weight" the particles of the local PHD filters such that the resulting new PHD approximates the AA of the local PHDs.

4.1. Motivation of AA Fusion

Calculating the AA of the local PHDs can be motivated as follows. Suppose sensor s wishes to estimate the number of targets in a region $\mathcal{R} \subseteq \mathbb{R}^d$, $N_k^{\mathcal{R}} = |X_k \cap \mathcal{R}|$, via the estimator (cf. (1)) $\hat{N}_{s,k}^{\mathcal{R},\mathrm{loc}} = \int_{\mathcal{R}} \hat{D}_{s,k}(\mathbf{x}|Z_{s,1:k})\mathrm{d}\mathbf{x}$. Since $Z_{s,1:k}$ is affected by clutter and missed detections, $\hat{N}_{s,k}^{\mathcal{R},\mathrm{loc}}$ may be quite different from $N_k^{\mathcal{R}}$. For example, if one target is in $\mathcal{R}$, i.e., $N_k^{\mathcal{R}} = 1$, sensor s may fail to detect that target, resulting in $\hat{N}_{s,k}^{\mathcal{R},\mathrm{loc}} \approx 0$; or if no target is in $\mathcal{R}$, i.e., $N_k^{\mathcal{R}} = 0$, a false alarm (clutter) at sensor s may lead to $\hat{N}_{s,k}^{\mathcal{R},\mathrm{loc}} \approx 1$. On the other hand, because the clutter and the missed detections at different sensors $s \in \mathcal{S}$ are not identical—in fact, they are assumed independent across the sensors—one can expect that the AA of the $\hat{N}_{s,k}^{\mathcal{R},\mathrm{loc}}$, i.e., $\hat{N}_{1:S,k}^{\mathcal{R}} \triangleq \sum_{s=1}^{S} \hat{N}_{s,k}^{\mathcal{R},\mathrm{loc}}/S$, is a more robust estimate of $N_k^{\mathcal{R}}$. This AA can be expressed as

$$\hat{N}_{1:S,k}^{\mathcal{R}} = \frac{1}{S} \sum_{s=1}^{S} \int_{\mathcal{R}} \hat{D}_{s,k}(\mathbf{x}|Z_{s,1:k})\mathrm{d}\mathbf{x} = \int_{\mathcal{R}} \hat{D}_k^{\mathrm{AA}}(\mathbf{x}|Z_{1:S,1:k})\mathrm{d}\mathbf{x},$$

with the AA of the local PHDs

$$\hat{D}_k^{\mathrm{AA}}(\mathbf{x}|Z_{1:S,1:k}) \triangleq \frac{1}{S} \sum_{s=1}^{S} \hat{D}_{s,k}(\mathbf{x}|Z_{s,1:k}). \tag{11}$$

Thus, $\hat{N}_{1:S,k}^{\mathcal{R}}$ is obtained by integrating the AA of the local PHDs over $\mathcal{R}$. This motivates a fusion of the local PHDs $\hat{D}_{s,k}(\mathbf{x}|Z_{s,1:k})$—thereby combining all the local measurements $Z_{s,1:k}$, $s = 1, \ldots, S$—by calculating the AA of the $\hat{D}_{s,k}(\mathbf{x}|Z_{s,1:k})$: we can expect that this compensates the effects of clutter and missed detections to some extent.

An interesting theoretical interpretation of AA fusion is given by the fact that AA fusion of the local posterior PHDs $\hat{D}_{s,k}(\mathbf{x}|Z_{s,1:k})$ minimizes the sum of the Kullback-Leibler divergences of the local posterior PHDs relative to the result of fusion, i.e.,

$$\hat{D}_k^{\mathrm{AA}}(\mathbf{x}|Z_{1:S,1:k}) = \arg\min_{\hat{D}(\mathbf{x})} \sum_{s=1}^{S} \mathrm{KLD}(\hat{D}_{s,k}(\mathbf{x}|Z_{s,1:k}) \| \hat{D}(\mathbf{x})). \tag{12}$$

This was shown in [21] for the analogous problem of fusing probability distributions. We note that, in a similar manner, GA fusion minimizes the sum of the Kullback-Leibler divergences of the result of fusion relative to the local posterior PHDs, i.e., $\mathrm{KLD}(\hat{D}_{s,k}(\mathbf{x}|Z_{s,1:k}) \| \hat{D}(\mathbf{x}))$ in (12) is formally replaced by $\mathrm{KLD}(\hat{D}(\mathbf{x}) \| \hat{D}_{s,k}(\mathbf{x}|Z_{s,1:k}))$ [43, 21, 4]. Furthermore, as discussed in [29], the results of both AA fusion and

GA fusion can be interpreted as specific instances of a Fréchet mean [28]. From (11), it is clear that AA fusion essentially aggregates the local PHDs, which tends to preserve their characteristics (such as modes). In GA fusion, on the other hand, certain local characteristics may be suppressed because the result of GA fusion is zero in a region of the state space even if only a single local PHD is zero in that region.

4.2. Outline of the Proposed PHD Fusion Scheme

In our approach, to reduce the amount of intersensor communication, the information exchanged between neighboring sensors consists of GM parameters rather than particles and weights. This necessitates conversions between particle representations and GM representations. The proposed AA-based fusion scheme thus consists of the following steps:

Step 1 Each sensor s converts its weighted particle set $\xi_{s,k} = \{(\mathbf{x}_{s,k}^{(j)}, w_{s,k}^{(j)})\}_{j=1}^{J_{s,k}}$ into a GM (see Section 5) and broadcasts the GM parameters to the neighboring sensors $r \in \mathcal{S}_s$.

Step 2 Each sensor s broadcasts its local cardinality estimate $W_{s,k}$ (see (4), (5)) to the neighboring sensors $r \in \mathcal{S}_s$.

Step 3 The GM parameters of each sensor s are fused with those received from the other sensors via a distributed dissemination/fusion scheme; see Section 6.

Step 4 The local cardinality estimate $W_{s,k}$ of each sensor s is fused with those received from the other sensors via a distributed dissemination/fusion scheme; see Section 8.1 [44].

Step 5 At each sensor s, the local particle weights $w_{s,k}^{(j)}$ are modified based on the fused GM parameters and the fused cardinality estimate; see Sections 7 and 8.1.

5. Particles–GM Conversion

In Step 1 of the above fusion scheme, the local weighted particle set $\xi_{s,k} = \{(\mathbf{x}_{s,k}^{(j)}, w_{s,k}^{(j)})\}_{j=1}^{J_{s,k}}$ is converted into a GM representation. Our conversion method differs from previous methods [45, 46, 47, 48, 12, 49, 50] in that it is based on the weight expansion in (7), i.e., $w_{s,k}^{(j)} = \sum_{\mathbf{z} \in Z_{s,k}^0} \omega_{s,k}^{(j)}(\mathbf{z})$. In our method, each of the $|Z_{s,k}^0| = M_{s,k} + 1$ extended measurements $\mathbf{z} \in Z_{s,k}^0 = \{\mathbf{z}_0, \mathbf{z}_{s,k}^{(1)}, \ldots, \mathbf{z}_{s,k}^{(M_{s,k})}\}$ potentially corresponds to one Gaussian component (GC) $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{s,k}(\mathbf{z}), \boldsymbol{\Sigma}_{s,k}(\mathbf{z}))$. Here, $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a Gaussian pdf with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The GC $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{s,k}(\mathbf{z}), \boldsymbol{\Sigma}_{s,k}(\mathbf{z}))$ is meant to represent the weighted particle set $\{(\mathbf{x}_{s,k}^{(j)}, \omega_{s,k}^{(j)}(\mathbf{z}))\}_{j=1}^{J_{s,k}}$. The mean vector $\boldsymbol{\mu}_{s,k}(\mathbf{z})$ and covariance matrix $\boldsymbol{\Sigma}_{s,k}(\mathbf{z})$ are derived from the respective weight components $\omega_{s,k}^{(j)}(\mathbf{z})$ and the particles $\mathbf{x}_{s,k}^{(j)}$ as

$$\boldsymbol{\mu}_{s,k}(\mathbf{z}) = \sum_{j=1}^{J_{s,k}} \bar{\omega}_{s,k}^{(j)}(\mathbf{z})\, \mathbf{x}_{s,k}^{(j)}, \tag{13}$$

$$\boldsymbol{\Sigma}_{s,k}(\mathbf{z}) = \sum_{j=1}^{J_{s,k}} \bar{\omega}_{s,k}^{(j)}(\mathbf{z}) (\mathbf{x}_{s,k}^{(j)} - \boldsymbol{\mu}_{s,k}(\mathbf{z})) (\mathbf{x}_{s,k}^{(j)} - \boldsymbol{\mu}_{s,k}(\mathbf{z}))^{\mathrm{T}}, \tag{14}$$

where $\bar{\omega}_{s,k}^{(j)}(\mathbf{z}) = \omega_{s,k}^{(j)}(\mathbf{z}) / \sum_{j'=1}^{J_{s,k}} \omega_{s,k}^{(j')}(\mathbf{z}) = \omega_{s,k}^{(j)}(\mathbf{z}) / \Omega_{s,k}(\mathbf{z})$ with $\omega_{s,k}^{(j)}(\mathbf{z})$ given by (8). In the overall GM-based PHD (briefly referred to as GM-PHD), the GC $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{s,k}(\mathbf{z}), \boldsymbol{\Sigma}_{s,k}(\mathbf{z}))$ is multiplied by the weight $\Omega_{s,k}(\mathbf{z}) =$

$\sum_{j=1}^{J_{s,k}} \omega_{s,k}^{(j)}(\mathbf{z})$ (see (9)). Thus, there is one weighted GC $\Omega_{s,k}(\mathbf{z})\,\mathcal{N}(\mathbf{x};\boldsymbol{\mu}_{s,k}(\mathbf{z}),\boldsymbol{\Sigma}_{s,k}(\mathbf{z}))$ for each measurement $\mathbf{z}\in Z_{s,k}^0$.

The overall GM-PHD is meant to represent the local weighted particle set $\xi_{s,k} = \{(\mathbf{x}_{s,k}^{(j)}, w_{s,k}^{(j)})\}_{j=1}^{J_{s,k}}$. Because $w_{s,k}^{(j)} = \sum_{\mathbf{z}\in Z_{s,k}^0} \omega_{s,k}^{(j)}(\mathbf{z})$, the overall GM-PHD is ideally taken to be the sum of all the weighted GCs, i.e.,

$$D_{s,k}^{\mathrm{GM,full}}(\mathbf{x}) \triangleq \sum_{\mathbf{z}\in Z_{s,k}^0} \Omega_{s,k}(\mathbf{z})\,\mathcal{N}(\mathbf{x};\boldsymbol{\mu}_{s,k}(\mathbf{z}),\boldsymbol{\Sigma}_{s,k}(\mathbf{z})). \tag{15}$$

This provides an approximate GM representation of $\hat{D}_{s,k}(\mathbf{x}|Z_{s,1:k})$. However, to further reduce the communication cost, we restrict the sum (15) to the GCs corresponding to "significant" measurements. (We note that a similar restriction was used previously for estimate extraction in [36, 37, 38].) The subset of significant measurements, $Z_{s,k}^{\mathrm{S}} \subseteq Z_{s,k}^0$, is defined as the set of those $\mathbf{z}\in Z_{s,k}^0$ for which $\Omega_{s,k}(\mathbf{z})$ in (9) is above a threshold T_Ω, where $0 < T_\Omega < 1$. In other words, the GM at sensor s contains a GC for $\mathbf{z} \in Z_{s,k}$ if the estimated probability that the measurement $\mathbf{z}$ originates from a target (given by $\Omega_{s,k}(\mathbf{z})$) is above T_Ω, and it contains a GC for $\mathbf{z}_0$ if the estimated number of missed detections (given by $\Omega_{s,k}(\mathbf{z}_0)$) is above T_Ω. Thus, the local GM-PHD is taken to be

$$D_{s,k}^{\mathrm{GM}}(\mathbf{x}) \triangleq \sum_{\mathbf{z}\in Z_{s,k}^{\mathrm{S}}} \Omega_{s,k}(\mathbf{z})\,\mathcal{N}(\mathbf{x};\boldsymbol{\mu}_{s,k}(\mathbf{z}),\boldsymbol{\Sigma}_{s,k}(\mathbf{z})). \tag{16}$$

This can be interpreted as the GM-PHD corresponding to the particle set $\{(\mathbf{x}_{s,k}^{(j)}, \breve{w}_{s,k}^{(j)})\}_{j=1}^{J_{s,k}}$ whose weights $\breve{w}_{s,k}^{(j)}$ are defined by summing the $\omega_{s,k}^{(j)}(\mathbf{z})$ only over the significant measurements, i.e., $\breve{w}_{s,k}^{(j)} = \sum_{\mathbf{z}\in Z_{s,k}^{\mathrm{S}}} \omega_{s,k}^{(j)}(\mathbf{z})$. We note that an alternative definition of a significant measurement subset $Z_{s,k}^{\mathrm{S}}$ and, thus, of $D_{s,k}^{\mathrm{GM}}(\mathbf{x})$ is to choose the $N_\Omega \triangleq \mathrm{round}\{W_{s,k}\}$ GCs with the largest $\Omega_{s,k}(\mathbf{z})$, $\mathbf{z}\in Z_{s,k}^0$. Here, $W_{s,k} = \sum_{\mathbf{z}\in Z_{s,k}^0} \Omega_{s,k}(\mathbf{z})$ according to (10), and we recall from (5) that $W_{s,k}$ approximates the MMSE estimate $\hat{N}_{s,k}^{\mathrm{MMSE}}$.

The suppression of GCs in the local GM-PHD $D_{s,k}^{\mathrm{GM}}(\mathbf{x})$ in (16) is motivated by the notion that "insignificant" measurements are likely to be false alarms (clutter). However, if an insignificant measurement is not a false alarm after all (and, thus, omitting it would lead to a missed detection), we can expect that it is not suppressed at most of the other sensors, and hence the erroneous suppression at sensor s will be compensated by the subsequent AA fusion. This is an advantage of AA fusion over GA fusion in dealing with missed detections of local filters [32].

Furthermore, because $D_{s,k}^{\mathrm{GM}}(\mathbf{x})$ only contains some of the GCs of $D_{s,k}^{\mathrm{GM,full}}(\mathbf{x})$, we can expect that also the result of AA fusion of all the $D_{s,k}^{\mathrm{GM}}(\mathbf{x})$ for $s = 1,\ldots,S$ will contain fewer GC components than if we fused all the $D_{s,k}^{\mathrm{GM,full}}(\mathbf{x})$. This principle, which was referred to as partial consensus in [10], counteracts the fact that AA fusion preserves all GC components [17] and thus is not able to remove false alarms [32]. As we will show in Section 10.2.4, the significance-based GC suppression performed in (16) allows the resulting distributed PHD filter to perform well also in strong clutter. Thus, we conclude that AA fusion with significance-based GC suppression is able to suppress many of the false alarms.

The GM parameter set underlying the local GM-PHD $D_{s,k}^{\mathrm{GM}}(\mathbf{x})$ in (16) is

$$\mathcal{G}_{s,k} \triangleq \{(\Omega_{s,k}(\mathbf{z}),\boldsymbol{\mu}_{s,k}(\mathbf{z}),\boldsymbol{\Sigma}_{s,k}(\mathbf{z}))\}_{\mathbf{z}\in Z_{s,k}^{\mathrm{S}}}. \tag{17}$$

All the further operations of our distributed PHD filter are based on $\mathcal{G}_{s,k}$; the GM-PHD $D_{s,k}^{\mathrm{GM}}(\mathbf{x})$ itself is never calculated. These further operations comprise a distributed fusion of the local GM parameter sets and of the local cardinality estimates, the conversion of the fused GM representations into particle representations, a

scaling of the particle weights, and the calculation of state estimates. A detailed presentation of these steps will be given in Sections 6–9.

6. Two GM Dissemination/Fusion Schemes

Once the local GM parameter sets $\mathcal{G}_{s,k}$ are available at the respective sensors s, they are disseminated and fused via a distributed scheme. The goal of this scheme is to obtain, at each sensor s, a GM parameter set that approximately corresponds to the AA of all the local GM-PHDs,

$$\bar{D}_k^{\mathrm{GM}}(\mathbf{x}) \triangleq \frac{1}{S}\sum_{s=1}^{S} D_{s,k}^{\mathrm{GM}}(\mathbf{x}). \tag{18}$$

Note that this equals (11) except that $\hat{D}_{s,k}(\mathbf{x}|Z_{s,1:k})$ is replaced by $D_{s,k}^{\mathrm{GM}}(\mathbf{x})$. Next, we discuss two alternative schemes for disseminating and fusing the local GM parameter sets.

6.1. GM Flooding

In the flooding scheme [49], each sensor s first broadcasts to its neighbors $r \in \mathcal{S}_s$ its GM parameter set $\mathcal{G}_{s,k} = \{(\Omega_{s,k}(\mathbf{z}), \boldsymbol{\mu}_{s,k}(\mathbf{z}), \boldsymbol{\Sigma}_{s,k}(\mathbf{z}))\}_{\mathbf{z}\in Z_{s,k}^{\mathrm{S}}}$ and receives their GM parameter sets $\mathcal{G}_{r,k} = \{(\Omega_{r,k}(\mathbf{z}), \boldsymbol{\mu}_{r,k}(\mathbf{z}), \boldsymbol{\Sigma}_{r,k}(\mathbf{z}))\}_{\mathbf{z}\in Z_{r,k}^{\mathrm{S}}}$, $r \in \mathcal{S}_s$. Each sensor then augments its own GM parameter set $\mathcal{G}_{s,k}$ by the neighbor GM parameter sets $\mathcal{G}_{r,k}$, $r \in \mathcal{S}_s$, resulting in $\mathcal{G}_{s,k}^{\mathrm{F}[1]} = \bigcup_{r\in\{s\}\cup\mathcal{S}_s} \mathcal{G}_{r,k}$. In the subsequent flooding iteration $i \in \{2, 3, \ldots\}$, each sensor s broadcasts to its neighbors the augmented set $\mathcal{G}_{s,k}^{\mathrm{F}[i-1]}$ with the exception of the elements already broadcast (the sensor keeps track of all the elements it already broadcast [49]) and receives the new elements of the neighbors' augmented sets $\mathcal{G}_{r,k}^{\mathrm{F}[i-1]}$. This results in the new augmented set

$$\mathcal{G}_{s,k}^{\mathrm{F}[i]} = \bigcup_{r\in\{s\}\cup\mathcal{S}_s} \mathcal{G}_{r,k}^{\mathrm{F}[i-1]}. \tag{19}$$

This recursion is initialized with $\mathcal{G}_{s,k}^{\mathrm{F}[0]} = \mathcal{G}_{s,k}$.

After the final flooding iteration $i = I$ (the choice of I will be discussed in Section 9.1), the augmented parameter set at sensor s is equal to

$$\mathcal{G}_{s,k}^{\mathrm{F}[I]} = \bigcup_{r\in\mathcal{S}_s^{[I]}} \mathcal{G}_{r,k}, \tag{20}$$

where $\mathcal{S}_s^{[I]} \subseteq \{1, 2, \ldots, S\}$ denotes the set of all those sensors that are at most I hops away from sensor s, including sensor s itself. At this point, sensor s would be able to calculate the AA of all the GM-PHDs whose GM parameters are contained in $\mathcal{G}_{s,k}^{[I]}$, i.e.,

$$D_{s,k}^{\mathrm{GM}[I]}(\mathbf{x}) = \frac{1}{\left|\mathcal{S}_s^{[I]}\right|}\sum_{r\in\mathcal{S}_s^{[I]}} D_{r,k}^{\mathrm{GM}}(\mathbf{x}). \tag{21}$$

If $I \geq R$, where R is the network diameter [51, 49], then $\mathcal{G}_{s,k}^{\mathrm{F}[I]}$ contains the GM parameters of all the sensors, and thus $D_{s,k}^{\mathrm{GM}[I]}(\mathbf{x})$ equals the total GM-PHD average $\bar{D}_k^{\mathrm{GM}}(\mathbf{x})$ in (18). (This presupposes that the sensor network is connected, which we assumed in Section 2.) If $I < R$, then $D_{s,k}^{\mathrm{GM}[I]}(\mathbf{x})$ provides only an approximation of $\bar{D}_k^{\mathrm{GM}}(\mathbf{x})$.

A drawback of the flooding scheme is that as the flooding iteration proceeds, the sets $\mathcal{G}_{s,k}^{\mathrm{F}[i]}$ grow in size since the GM parameters of additional sensors are included. Indeed, in iteration i, sensor s receives the GM parameters $\{\mathcal{G}_{r,k}\}_{r\in\Delta\mathcal{S}_s^{[i]}}$, where $\Delta\mathcal{S}_s^{[i]} \subseteq \{1, 2, \ldots, S\}$ comprises all sensors that are exactly i hops away from

sensor s; note that $\Delta\mathcal{S}_s^{[i]} = \mathcal{S}_s^{[i]} \setminus \mathcal{S}_s^{[i-1]}$. These GM parameters are added to the previous GM parameter set of sensor s, $\mathcal{G}_{s,k}^{\mathrm{F}[i-1]}$. Thus, Eq. (19) can be reformulated as

$$\mathcal{G}_{s,k}^{\mathrm{F}[i]} = \mathcal{G}_{s,k}^{\mathrm{F}[i-1]} \cup \bigcup_{r \in \Delta\mathcal{S}_s^{[i]}} \mathcal{G}_{r,k}. \tag{22}$$

The total number of real values that have to be broadcast in iteration i by all the sensors in the network is equal to the number of real values needed to specify all the elements of the set $\bigcup_{s=1}^{S} \bigcup_{r \in \Delta\mathcal{S}_s^{[i]}} \mathcal{G}_{r,k}$.

6.2. GM Average Consensus

To limit the growth of the GM parameter sets and to reduce the communication cost, we may emulate a part of the averaging in (21) in each iteration i. To this end, we consider a formal application of the average consensus algorithm [52, 51] to the local GM-PHDs. According to that algorithm, the iterated GM-PHD at sensor s—denoted by $D_{s,k}^{\mathrm{cons}[i]}(\mathbf{x})$—would be updated in iteration i as

$$D_{s,k}^{\mathrm{cons}[i]}(\mathbf{x}) = \sum_{r \in \{s\} \cup \mathcal{S}_s} \alpha_{s,r} D_{r,k}^{\mathrm{cons}[i-1]}(\mathbf{x}), \tag{23}$$

with appropriately chosen weights $\alpha_{s,r}$, where $s, r \in \{1, 2, \ldots, S\}$. A popular choice is given by the Metropolis weights [52] defined as $\alpha_{s,r} = 1/(1 + \max(|\mathcal{S}_r|, |\mathcal{S}_s|))$ if $r \neq s$ and $\alpha_{s,s} = 1 - \sum_{r \in \mathcal{S}_s} \alpha_{s,r}$. The recursion (23) is initialized as $D_{s,k}^{\mathrm{cons}[0]}(\mathbf{x}) = D_{s,k}^{\mathrm{GM}}(\mathbf{x})$ (see (16)). Since the network is connected, $D_{s,k}^{\mathrm{cons}[i]}(\mathbf{x})$ is guaranteed to converge for $i \to \infty$ to the total GM-PHD average $\bar{D}_k^{\mathrm{GM}}(\mathbf{x})$ in (18) [52]. For a finite number I of iterations, $D_{s,k}^{\mathrm{cons}[I]}(\mathbf{x})$ provides only an approximation of $\bar{D}_k^{\mathrm{GM}}(\mathbf{x})$.

A direct implementation of the update (23) is impossible because the iterated GM-PHDs $D_{s,k}^{\mathrm{cons}[i]}(\mathbf{x})$ are functions, rather than numbers. Therefore, we will emulate (23) through operations involving the GM parameters of the iterated local GM-PHDs $D_{s,k}^{\mathrm{cons}[i]}(\mathbf{x})$ and $D_{r,k}^{\mathrm{cons}[i-1]}(\mathbf{x})$, $r \in \{s\} \cup \mathcal{S}_s$ involved in (23). First, as in the flooding scheme discussed in Section 6.1, each sensor s broadcasts to its neighbors $r \in \mathcal{S}_s$ its GM parameter set $\mathcal{G}_{s,k} = \{(\Omega_{s,k}(\mathbf{z}), \boldsymbol{\mu}_{s,k}(\mathbf{z}), \boldsymbol{\Sigma}_{s,k}(\mathbf{z}))\}_{\mathbf{z} \in Z_{s,k}^{\mathrm{S}}}$ (see (17)) and receives their GM parameter sets $\mathcal{G}_{r,k}$. Then, sensor s scales each GM weight $\Omega_{r,k}(\mathbf{z})$ with the corresponding consensus weight $\alpha_{s,r}$, resulting in the scaled weights $\Omega_{s,r,k}^{(\alpha)}(\mathbf{z}) \triangleq \alpha_{s,r} \Omega_{r,k}(\mathbf{z})$, for $\mathbf{z} \in Z_{r,k}^{\mathrm{S}}$, $r \in \{s\} \cup \mathcal{S}_s$. Thus, sensor s now disposes of the "scaled GM parameter sets"

$$\mathcal{G}_{s,r,k}^{(\alpha)} \triangleq \{(\Omega_{s,r,k}^{(\alpha)}(\mathbf{z}), \boldsymbol{\mu}_{r,k}(\mathbf{z}), \boldsymbol{\Sigma}_{r,k}(\mathbf{z}))\}_{\mathbf{z} \in Z_{r,k}^{\mathrm{S}}},$$

for all $r \in \{s\} \cup \mathcal{S}_s$. The GM-PHD generated in analogy to (16) from the union of all these GM parameter sets, $\mathcal{G}_{s,k}^{\cup} \triangleq \bigcup_{r \in \{s\} \cup \mathcal{S}_s} \mathcal{G}_{s,r,k}^{(\alpha)}$, would be

$$\begin{aligned} D_{s,k}^{\mathrm{GM},\cup}(\mathbf{x}) &\triangleq \sum_{r \in \{s\} \cup \mathcal{S}_s} \sum_{\mathbf{z} \in Z_{r,k}^{\mathrm{S}}} \Omega_{s,r,k}^{(\alpha)}(\mathbf{z}) \, \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{r,k}(\mathbf{z}), \boldsymbol{\Sigma}_{r,k}(\mathbf{z})) \\ &= \sum_{r \in \{s\} \cup \mathcal{S}_s} \alpha_{s,r} \sum_{\mathbf{z} \in Z_{r,k}^{\mathrm{S}}} \Omega_{r,k}(\mathbf{z}) \, \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{r,k}(\mathbf{z}), \boldsymbol{\Sigma}_{r,k}(\mathbf{z})) \\ &= \sum_{r \in \{s\} \cup \mathcal{S}_s} \alpha_{s,r} D_{r,k}^{\mathrm{GM}}(\mathbf{x}), \end{aligned} \tag{24}$$

where (16) was used in the last step. A comparison with (23) shows that we have emulated the first GM-PHD average consensus iteration ($i = 1$) by operating solely at the level of the GM parameters [10]. Note that $D_{s,k}^{\mathrm{GM},\cup}(\mathbf{x})$ (or any other PHD) is not actually computed by the proposed algorithm.

Just as the flooding scheme, this scheme suffers from the fact that the fused GM parameter set at sensor s, $\mathcal{G}_{s,k}^{\cup} = \bigcup_{r\in\{s\}\cup\mathcal{S}_s} \mathcal{G}_{s,r,k}^{(\alpha)}$, is much larger than the original GM parameter set $\mathcal{G}_{s,k}$. Therefore, we apply *mixture reduction* [53, 54, 10] to $\mathcal{G}_{s,k}^{\cup}$, resulting in a reduced GM parameter set $\mathcal{G}_{s,k}^{[1]} \triangleq \{(\Omega_{s,k,\ell}^{[1]}, \boldsymbol{\mu}_{s,k,\ell}^{[1]}, \boldsymbol{\Sigma}_{s,k,\ell}^{[1]})\}_{\ell\in L_{s,k}^{[1]}}$, where $L_{s,k}^{[1]}$ is some reduced index set. The GM-PHD corresponding to $\mathcal{G}_{s,k}^{[1]}$, i.e.,

$$D_{s,k}^{\mathrm{GM}[1]}(\mathbf{x}) \triangleq \sum_{\ell\in L_{s,k}^{[1]}} \Omega_{s,k,\ell}^{[1]}\, \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{s,k,\ell}^{[1]}, \boldsymbol{\Sigma}_{s,k,\ell}^{[1]}), \tag{25}$$

is then only an approximation of $D_{s,k}^{\mathrm{GM},\cup}(\mathbf{x})$. Mixture reduction usually consists of merging GCs that are "close" with respect to an appropriate metric, and pruning GCs with small weights. In our case, the weights are not small because they survived the thresholding performed in Section 5, and thus we only perform a merging operation.

These union and merging operations are repeated in all the further iterations. In iteration $i \in \{2, 3, \ldots\}$, sensor s broadcasts to its neighbors the set $\mathcal{G}_{s,k}^{[i-1]} = \{(\Omega_{s,k,\ell}^{[i-1]}, \boldsymbol{\mu}_{s,k,\ell}^{[i-1]}, \boldsymbol{\Sigma}_{s,k,\ell}^{[i-1]})\}_{\ell\in L_{s,k}^{[i-1]}}$ and receives their sets $\mathcal{G}_{r,k}^{[i-1]}$, $r\in\mathcal{S}_s$. It then scales each GM weight $\Omega_{r,k,\ell}^{[i-1]}$, $\ell\in L_{r,k}^{[i-1]}$, $r\in\{s\}\cup\mathcal{S}_s$ with the corresponding consensus weight $\alpha_{s,r}$. This results in the "scaled GM parameter sets"

$$\mathcal{G}_{s,r,k}^{[i-1](\alpha)} \triangleq \{(\Omega_{s,r,k,\ell}^{[i-1](\alpha)}, \boldsymbol{\mu}_{r,k,\ell}^{[i-1]}, \boldsymbol{\Sigma}_{r,k,\ell}^{[i-1]})\}_{\ell\in L_{r,k}^{[i-1]}}, \quad r \in \{s\}\cup\mathcal{S}_s,$$

with $\Omega_{s,r,k,\ell}^{[i-1](\alpha)} \triangleq \alpha_{s,r}\Omega_{r,k,\ell}^{[i-1]}$. Let $D_{s,k}^{\mathrm{GM}[i-1],\cup}(\mathbf{x})$ denote the GM-PHD corresponding to the union of all these GM parameter sets, $\mathcal{G}_{s,k}^{[i-1],\cup} \triangleq \bigcup_{r\in\{s\}\cup\mathcal{S}_s} \mathcal{G}_{s,r,k}^{[i-1](\alpha)}$, i.e.,

$$D_{s,k}^{\mathrm{GM}[i-1],\cup}(\mathbf{x}) \triangleq \sum_{r\in\{s\}\cup\mathcal{S}_s} \sum_{\ell\in L_{r,k}^{[i-1]}} \Omega_{s,r,k,\ell}^{[i-1](\alpha)}\, \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{r,k,\ell}^{[i-1]}, \boldsymbol{\Sigma}_{r,k,\ell}^{[i-1]}).$$

Using (25) with obvious modifications, i.e., $D_{s,k}^{\mathrm{GM}[i-1]}(\mathbf{x}) = \sum_{\ell\in L_{s,k}^{[i-1]}} \Omega_{s,k,\ell}^{[i-1]}\, \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{s,k,\ell}^{[i-1]}, \boldsymbol{\Sigma}_{s,k,\ell}^{[i-1]})$, we obtain (cf. (24))

$$D_{s,k}^{\mathrm{GM}[i-1],\cup}(\mathbf{x}) = \sum_{r\in\{s\}\cup\mathcal{S}_s} \alpha_{s,r} D_{r,k}^{\mathrm{GM}[i-1]}(\mathbf{x}). \tag{26}$$

Hence, we have emulated the GM-PHD average consensus iteration (23) by operating at the level of the GM parameters. Finally, a merging step reduces $\mathcal{G}_{s,k}^{[i-1],\cup}$ to a smaller GM parameter set

$$\mathcal{G}_{s,k}^{[i]} \triangleq \{(\Omega_{s,k,\ell}^{[i]}, \boldsymbol{\mu}_{s,k,\ell}^{[i]}, \boldsymbol{\Sigma}_{s,k,\ell}^{[i]})\}_{\ell\in L_{s,k}^{[i]}}.$$

The GM-PHD corresponding to $\mathcal{G}_{s,k}^{[i]}$, i.e.,

$$D_{s,k}^{\mathrm{GM}[i]}(\mathbf{x}) \triangleq \sum_{\ell\in L_{s,k}^{[i]}} \Omega_{s,k,\ell}^{[i]}\, \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{s,k,\ell}^{[i]}, \boldsymbol{\Sigma}_{s,k,\ell}^{[i]}), \tag{27}$$

approximates $D_{s,k}^{\mathrm{GM}[i-1],\cup}(\mathbf{x})$ in (26). The recursion $\mathcal{G}_{s,k}^{[i-1]} \to \mathcal{G}_{s,k}^{[i]}$ described above is initialized with $\mathcal{G}_{s,k}^{[0]} = \mathcal{G}_{s,k}$.

Thus, after I iterations, we have converted the original local GM parameter set $\mathcal{G}_{s,k}$ into a fused GM parameter set $\mathcal{G}_{s,k}^{[I]}$ that approximately emulates I average consensus iterations (23). The choice of I will be discussed in Section 9.1. In conclusion, we have developed an approximate implementation of the GM-PHD average consensus scheme (23) that operates at the level of the GM parameters. Note that here—in contrast to the distributed flooding scheme discussed in Section 6.1—the iterated GM parameter sets $\mathcal{G}_{s,k}^{[i]}$ do not systematically grow with progressing iteration i. Furthermore, our experimental results reported in Section 10 suggest that the proposed GM average consensus scheme with GC merging can outperform the GM flooding scheme in terms of tracking accuracy.

7. IS Method for GM–Particles Conversion

The dissemination/fusion schemes discussed in the previous section effectively provide each sensor s with a fused GM-PHD $D_{s,k}^{\mathrm{GM}[I]}(\mathbf{x})$, which is given by (21) if the GM flooding scheme of Section 6.1 is used and by (27) (with i replaced by I) if the GM average consensus scheme of Section 6.2 is used. (We say "effectively" because $D_{s,k}^{\mathrm{GM}[I]}(\mathbf{x})$ is not actually calculated.) In what follows, we will denote by

$$\mathcal{G}_{s,k}^{[I]} \triangleq \{(\Omega_{s,k,\ell}^{[I]}, \boldsymbol{\mu}_{s,k,\ell}^{[I]}, \boldsymbol{\Sigma}_{s,k,\ell}^{[I]})\}_{\ell \in L_{s,k}^{[I]}} \tag{28}$$

the set of GM parameters involved in $D_{s,k}^{\mathrm{GM}[I]}(\mathbf{x})$, i.e., we have

$$D_{s,k}^{\mathrm{GM}[I]}(\mathbf{x}) = \sum_{\ell \in L_{s,k}^{[I]}} \Omega_{s,k,\ell}^{[I]}\, \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{s,k,\ell}^{[I]}, \boldsymbol{\Sigma}_{s,k,\ell}^{[I]}). \tag{29}$$

Here, in the case of GM flooding, $\mathcal{G}_{s,k}^{[I]}$ is obtained from $\mathcal{G}_{s,k}^{\mathrm{F}[I]}$ in (20) by scaling all the weights in $\mathcal{G}_{s,k}^{\mathrm{F}[I]}$ with the factor $1/|\mathcal{S}_s^{[I]}|$; this accounts for the factor $1/|\mathcal{S}_s^{[I]}|$ in (21).

In order to use the fused GM-PHD $D_{s,k}^{\mathrm{GM}[I]}(\mathbf{x})$ in the local particle-PHD filter at sensor s, it is necessary to find a particle representation of $D_{s,k}^{\mathrm{GM}[I]}(\mathbf{x})$. The standard method is to sample directly from $D_{s,k}^{\mathrm{GM}[I]}(\mathbf{x})$. However, we here propose a method based on the importance sampling (IS) principle [39, Ch. 3.3], which will be seen in Section 9.1 to enable a parallelization of filtering and fusion operations. We start by recalling from Section 3 that the local PHD filter propagates over time k a weighted particle set $\xi_{s,k} = \{(\mathbf{x}_{s,k}^{(j)}, w_{s,k}^{(j)})\}_{j=1}^{J_{s,k}}$ providing an approximate representation of $\hat{D}_{s,k}(\mathbf{x}|Z_{s,1:k})$. Let us now consider an alternative particle representation of $\hat{D}_{s,k}(\mathbf{x}|Z_{s,1:k})$ using a uniformly weighted particle set $\{(\tilde{\mathbf{x}}_{s,k}^{(j)}, c_{s,k})\}_{j=1}^{\tilde{J}_{s,k}}$. Here, the number of uniformly weighted particles is chosen as

$$\tilde{J}_{s,k} = \mathrm{round}\{N_{\mathrm{p}} W_{s,k}\}, \tag{30}$$

where $N_{\mathrm{p}} \in \mathbb{N}$ is a parameter specifying the number of particles assigned to each potential target, as discussed in [2, Sec. III.C], and, as before (see (4)), $W_{s,k}$ is the sum of the original weights $w_{s,k}^{(j)}$. Furthermore, the weight $c_{s,k}$—identical for all j—is

$$c_{s,k} = \frac{W_{s,k}}{\tilde{J}_{s,k}}.$$

The new particles $\tilde{\mathbf{x}}_k^{(j)}$ are obtained from the original weighted particle set $\xi_{s,k}$ through resampling, which means that particles with large weights are replicated whereas those with small weights are removed [55]. As such, each resampled particle $\tilde{\mathbf{x}}_{s,k}^{(j)}$ equals one of the original particles, $\mathbf{x}_{s,k}^{(j')}$, where j' is uniquely determined by j. Note that some of the $\tilde{\mathbf{x}}_{s,k}^{(j)}$ are identical due to the replication. Let $N_{s,k}^{(j')}$ denote the number of times particle $\mathbf{x}_{s,k}^{(j')}$ is resampled (replicated). To ensure unbiased resampling, we require that the expectation of $N_{s,k}^{(j')}$ given $\xi_{s,k} = \{(\mathbf{x}_{s,k}^{(j)}, w_{s,k}^{(j)})\}_{j=1}^{J_{s,k}}$ is N_{p} times $w_{s,k}^{(j')}$ [55], i.e.,

$$\mathrm{E}[N_{s,k}^{(j')} | \xi_{s,k}] = N_{\mathrm{p}}\, w_{s,k}^{(j')}. \tag{31}$$

As verified in the Appendix, this can be achieved approximately by choosing a new particle $\tilde{\mathbf{x}}_{s,k}^{(j)}$ equal to $\mathbf{x}_{s,k}^{(j')}$ with probability

$$P_{j'} \triangleq \Pr[\tilde{\mathbf{x}}_{s,k}^{(j)} = \mathbf{x}_{s,k}^{(j')} | \xi_{s,k}] = \frac{w_{s,k}^{(j')}}{W_{s,k}}. \tag{32}$$

The resampled particle set $\{(\tilde{\mathbf{x}}_{s,k}^{(j)}, c_{s,k} = W_{s,k}/\tilde{J}_{s,k})\}_{j=1}^{\tilde{J}_{s,k}}$ represents $\hat{D}_{s,k}(\mathbf{x}|Z_{s,1:k})$. However, based on the IS principle [39, Ch. 3.3], we can also use $\{\tilde{\mathbf{x}}_{s,k}^{(j)}\}_{j=1}^{\tilde{J}_{s,k}}$ to represent the fused GM-PHD[1] $D_{s,k}^{\mathrm{GM}[I]}(\mathbf{x})$ in (29), if only the weight associated with $\tilde{\mathbf{x}}_{s,k}^{(j)} = \mathbf{x}_{s,k}^{(j')}$ is chosen as

$$\tilde{w}_{s,k}^{(j)} = \frac{D_{s,k}^{\mathrm{GM}[I]}(\tilde{\mathbf{x}}_{s,k}^{(j)})}{P_{j'}} = \frac{W_{s,k} D_{s,k}^{\mathrm{GM}[I]}(\tilde{\mathbf{x}}_{s,k}^{(j)})}{w_{s,k}^{(j')}}, \tag{33}$$

where, from (29),

$$D_{s,k}^{\mathrm{GM}[I]}(\tilde{\mathbf{x}}_{s,k}^{(j)}) = \sum_{\ell \in L_{s,k}^{[I]}} \Omega_{s,k,\ell}^{[I]} \, \mathcal{N}(\tilde{\mathbf{x}}_{s,k}^{(j)}; \boldsymbol{\mu}_{s,k,\ell}^{[I]}, \boldsymbol{\Sigma}_{s,k,\ell}^{[I]}), \tag{34}$$

with $\tilde{\mathbf{x}}_{s,k}^{(j)} = \mathbf{x}_{s,k}^{(j')}$. Hereafter, we use $\{(\tilde{\mathbf{x}}_{s,k}^{(j)}, \tilde{w}_{s,k}^{(j)})\}_{j=1}^{\tilde{J}_{s,k}}$ to represent $D_{s,k}^{\mathrm{GM}[I]}(\mathbf{x})$. The particle set conversion $\xi_{s,k} \to \{(\tilde{\mathbf{x}}_{s,k}^{(j)}, \tilde{w}_{s,k}^{(j)})\}_{j=1}^{\tilde{J}_{s,k}}$ developed above constitutes a particle implementation of the PHD fusion conversion $\hat{D}_{s,k}(\mathbf{x}|Z_{s,1:k}) \to D_{s,k}^{\mathrm{GM}[I]}(\mathbf{x})$.

8. Cardinality Averaging and State Estimation

In this section, we discuss two final stages of our distributed PHD filtering method.

8.1. AA-based Cardinality Averaging

By (5), $W_{s,k} = \sum_{\mathbf{z} \in Z_{s,k}^0} \Omega_{s,k}(\mathbf{z})$ (see (10)) provides an estimate of the cardinality $N_k = |X_k|$. However, in our particles–GM conversion method (see Section 5), $Z_{s,k}^0$ was replaced by the subset $Z_{s,k}^{\mathrm{S}}$, and consequently $\sum_{\mathbf{z} \in Z_{s,k}^0} \Omega_{s,k}(\mathbf{z})$ is replaced by $\sum_{\mathbf{z} \in Z_{s,k}^{\mathrm{S}}} \Omega_{s,k}(\mathbf{z}) \leq \sum_{\mathbf{z} \in Z_{s,k}^0} \Omega_{s,k}(\mathbf{z})$. This implies that the fused GM-PHD $D_{s,k}^{\mathrm{GM}[I]}(\mathbf{x})$ in (29) and the associated weights $\tilde{w}_{s,k}^{(j)}$ in (33) will both underestimate the cardinality N_k, in the sense that, typically, $\int_{\mathbb{R}^d} D_{s,k}^{\mathrm{GM}[I]}(\mathbf{x}) \mathrm{d}\mathbf{x} < N_k$ and $\sum_{j=1}^{\tilde{J}_{s,k}} \tilde{w}_{s,k}^{(j)} < N_k$.

This "cardinality bias" can be compensated by a suitable scaling of the weights $\tilde{w}_{s,k}^{(j)}$. In our method (see Steps 4 and 5 in Section 4), following [44], this scaling is based on the original—"correct"— local cardinality estimates $W_{s,k}$, which are averaged over all sensors s to smooth out sensor-specific errors. That is, we attempt to calculate the AA of all the local cardinality estimates, $\overline{W}_k \triangleq \sum_{s=1}^{S} W_{s,k}/S$, and use the result to scale the $\tilde{w}_{s,k}^{(j)}$. Note that $\overline{W}_k = \int_{\mathbb{R}^d} \hat{D}_k^{\mathrm{AA}}(\mathbf{x}|Z_{1:S,1:k}) \mathrm{d}\mathbf{x}$ with $\hat{D}_k^{\mathrm{AA}}(\mathbf{x}|Z_{1:S,1:k}) = \sum_{s=1}^{S} \hat{D}_{s,k}(\mathbf{x}|Z_{s,1:k})/S$ as defined in (11), which means that $\overline{W}_k$ is the cardinality estimate based on the AA of all the local PHDs $\hat{D}_{s,k}(\mathbf{x}|Z_{s,1:k})$.

For a distributed approximate calculation of $\overline{W}_k$, we can use flooding or average consensus on the $W_{s,k}$ (cf. Section 6) [44]. Let $W_{s,k}^{[I]}$ be the approximation of $\overline{W}_k$ obtained after I flooding or average consensus iterations. Then, the weights $\tilde{w}_{s,k}^{(j)}$ are scaled as

$$\bar{w}_{s,k}^{(j)} = \beta_{s,k} \tilde{w}_{s,k}^{(j)}, \quad j = 1, \ldots, \tilde{J}_{s,k}, \tag{35}$$

where, as derived in [44],

$$\beta_{s,k} = \frac{W_{s,k}^{[I]}}{\sum_{j=1}^{\tilde{J}_{s,k}} \tilde{w}_{s,k}^{(j)}}. \tag{36}$$

[1] This representation can be expected to be accurate only if the effective support of $D_{s,k}^{\mathrm{GM}[I]}(\mathbf{x})$ is contained in that of $\hat{D}_{s,k}(\mathbf{x}|Z_{s,1:k})$. This condition is satisfied for all s if the fields of view of all sensors are effectively equal. In the opposite case, one has to expect a performance loss compared to the standard method of sampling directly from $D_{s,k}^{\mathrm{GM}[I]}(\mathbf{x})$.

We then use $\{(\tilde{\mathbf{x}}_{s,k}^{(j)}, \bar{w}_{s,k}^{(j)})\}_{j=1}^{\tilde{J}_{s,k}}$ as the final particle representation of the fused PHD $D_{s,k}^{\mathrm{GM}[I]}(\mathbf{x})$. In the local PHD filter at sensor s, $\{(\tilde{\mathbf{x}}_{s,k}^{(j)}, \bar{w}_{s,k}^{(j)})\}_{j=1}^{\tilde{J}_{s,k}}$ replaces the original particle representation $\xi_{s,k}$, i.e., it is used instead of $\xi_{s,k}$ in the next prediction step. We note that an accurate cardinality estimate is also crucial for target state estimation, as explained next.

8.2. Target State Estimation

At each sensor s and time k, estimates of the target states $\mathbf{x}_k^{(\nu)}$ are calculated as follows. First, an estimate of the number of targets is formed as $\hat{N}_{s,k} \triangleq \mathrm{round}\{W_{s,k}^{[I]}\}$, where $W_{s,k}^{[I]}$ is the result of the cardinality averaging scheme discussed above. Then, the means of the $\hat{N}_{s,k}$ GCs with the largest weights $\Omega_{s,k,\ell}^{[I]}$ are used as estimates of the target states.[2] Note that this target state estimation operation is performed locally at sensor s.

9. Algorithm Summary, Parallelization, and Communication Cost

9.1. Algorithm Summary and Parallelization

A summary of the proposed distributed PHD filter algorithm is given in Algorithm 1. Two noteworthy aspects are that (i) the filtering operations 1 and 2 do not require or change the previous particle weights, and (ii) the fusion-related operations 8–15 do not change the current particles. As a consequence, the filtering operations 1 and 2 for time $k+1$ can be carried out in parallel (simultaneously) with the fusion-related operations 8–15 for time k. More specifically, operations 1 and 2 for time $k+1$ can be carried out as soon as operation 7 for time k is done; they do not need to wait for the results of operations 8–17. Also, operations 8–10 for time k can be performed in parallel with operations 5–7 for time k. In summary, the filtering operations 1 and 2 for time $k+1$ and the filtering operations 5–7 for time k can be performed in parallel with the fusion-related operations 8–15 for time k. Since operation 2 (including calculation of $g_{s,k}(\mathbf{z}|\mathbf{x}_{s,k}^{(j)})$) and operation 7 (resampling) are the most computationally intensive filtering operations, a large degree of parallelization is possible. A timing diagram illustrating the scheduling and parallelization of the various operations is given in Fig. 1.

This parallelization, which is enabled by our IS method for GM–particles conversion, is an important advantage of the proposed distributed PHD filter algorithm. Indeed, with most other distributed PHD filtering algorithms, the filtering operations can only be scheduled before or after the dissemination/fusion operations. Because the time duration Δ of one filtering step (corresponding to one time step $k \to k+1$) is limited by the time between two sensing scans, this serial schedule implies a strong limitation of the number I of dissemination/fusion iterations that can be carried out in each filtering step. More specifically, for any distributed filtering algorithm, the maximum possible value of I is

$$I_{\max} = \left\lfloor \frac{\Delta - t_{\mathrm{filt}} - t_{\mathrm{inter}}}{\tau} \right\rfloor. \tag{37}$$

Here, t_{filt} is the total time duration of all the filtering operations that cannot be carried out in parallel with the dissemination/fusion iterations; t_{inter} is the time required by operations interfacing the dissemination/fusion scheme with the local filtering (preparing data to be communicated, inserting the communicated data into

[2] An alternative method is to group all the GC means into $\hat{N}_{s,k}$ clusters and use the weighted average of the means of each cluster as a state estimate. However, this method is more complex and, moreover, did not perform better in our simulations.

Algorithm 1 Proposed distributed particle-PHD filter algorithm—operations performed at sensor s during time step k

Input: Previous particle set $\{(\mathbf{x}_{s,k-1}^{(j)}, w_{s,k-1}^{(j)})\}_{j=1}^{J_{s,k-1}}$; measurement set $Z_{s,k}$; number of newborn particles $L_{s,k}$.

Output: New particle set $\{(\tilde{\mathbf{x}}_{s,k}^{(j)}, \bar{w}_{s,k}^{(j)})\}_{j=1}^{\tilde{J}_{s,k}}$ (this particle set will be used as the input—see above—at the next time step $k+1$); target state estimates $\hat{\mathbf{x}}_{s,k}^{(\nu)}$, $\nu = 1, \ldots, \hat{N}_{s,k}$.

Operations:

Local filtering

1. For $j = 1, \ldots, J_{s,k}$, with $J_{s,k} = J_{s,k-1} + L_{s,k}$, draw particles $\mathbf{x}_{s,k}^{(j)}$ from proposal pdf $q_{s,k}(\mathbf{x}; \mathbf{x}_{s,k-1}^{(j)}, Z_{s,k})$ (if $j \in \{1, \ldots, J_{s,k-1}\}$) or $p_{s,k}(\mathbf{x}; Z_{s,k})$ (if $j \in \{J_{s,k-1}+1, \ldots, J_{s,k}\}$).
2. Evaluate $f_k(\mathbf{x}_{s,k}^{(j)}|\mathbf{x}_{s,k-1}^{(j)})$ and $q_{s,k}(\mathbf{x}_{s,k}^{(j)}; \mathbf{x}_{s,k-1}^{(j)}, Z_{s,k})$ for $j = 1, \ldots, J_{s,k-1}$; $\gamma_k(\mathbf{x}_{s,k}^{(j)})$ and $p_{s,k}(\mathbf{x}_{s,k}^{(j)}; Z_{s,k})$ for $j = J_{s,k-1}+1, \ldots, J_{s,k}$; $p_{s,k}^{\mathrm{D}}(\mathbf{x}_{s,k}^{(j)})$ for $j = 1, \ldots, J_{s,k}$; $g_{s,k}(\mathbf{z}|\mathbf{x}_{s,k}^{(j)})$ for $\mathbf{z} \in Z_{s,k}$ and $j = 1, \ldots, J_{s,k}$; and $\kappa_{s,k}(\mathbf{z})$ for $\mathbf{z} \in Z_{s,k}$.
3. Calculate $w_{s,k|k-1}^{(j)}$ for $j = 1, \ldots, J_{s,k}$ using (6).
4. Calculate $\omega_{s,k}^{(j)}(\mathbf{z})$ for $\mathbf{z} \in Z_{s,k}^0$ and $j = 1, \ldots, J_{s,k}$ using (8).
5. Calculate $w_{s,k}^{(j)}$ for $j = 1, \ldots, J_{s,k}$ using (7).
6. Calculate $W_{s,k}$ according to (4).
7. Resample $\{(\mathbf{x}_{s,k}^{(j)}, w_{s,k}^{(j)})\}_{j=1}^{J_{s,k}}$ to obtain a uniformly weighted particle set $\{(\tilde{\mathbf{x}}_{s,k}^{(j)}, c_{s,k})\}_{j=1}^{\tilde{J}_{s,k}}$, where $c_{s,k} = W_{s,k}/\tilde{J}_{s,k}$ with $\tilde{J}_{s,k} = \mathrm{round}\{N_{\mathrm{p}} W_{s,k}\}$. For $j = 1, \ldots, \tilde{J}_{s,k}$, store the weight $w_{s,k}^{(j')}$ of the particle $\mathbf{x}_{s,k}^{(j')}$ associated with $\tilde{\mathbf{x}}_{s,k}^{(j)}$.

Fusion

8. Calculate $\Omega_{s,k}(\mathbf{z})$ for $\mathbf{z} \in Z_{s,k}^0$ according to (9).
9. Determine the subset of significant measurements, $Z_{s,k}^{\mathrm{S}} \subseteq Z_{s,k}^0$, as the set of those $\mathbf{z} \in Z_{s,k}^0$ for which $\Omega_{s,k}(\mathbf{z}) > T_\Omega$.
10. For $\mathbf{z} \in Z_{s,k}^{\mathrm{S}}$, determine $\boldsymbol{\mu}_{s,k}(\mathbf{z})$ and $\boldsymbol{\Sigma}_{s,k}(\mathbf{z})$ according to (13) and (14), respectively.
11. Calculate the fused GM parameter set $\mathcal{G}_{s,k}^{[I]} = \{(\Omega_{s,k,\ell}^{[I]}, \boldsymbol{\mu}_{s,k,\ell}^{[I]}, \boldsymbol{\Sigma}_{s,k,\ell}^{[I]})\}_{\ell \in L_{s,k}^{[I]}}$ (cf. (28)) by performing I iterations of a distributed dissemination/fusion scheme as described in Section 6. This requires broadcasting data to sensors $r \in \mathcal{S}_s$.
12. Calculate the fused cardinality estimate $W_{s,k}^{[I]}$ by means of distributed cardinality averaging as described in Section 8.1. This requires broadcasting data to sensors $r \in \mathcal{S}_s$.
13. Calculate $D_{s,k}^{\mathrm{GM}[I]}(\tilde{\mathbf{x}}_{s,k}^{(j)})$ for $j = 1, \ldots, \tilde{J}_{s,k}$ using (34).
14. Calculate $\tilde{w}_{s,k}^{(j)}$ for $j = 1, \ldots, \tilde{J}_{s,k}$ using (33).
15. Calculate $\bar{w}_{s,k}^{(j)}$ for $j = 1, \ldots, \tilde{J}_{s,k}$ using (35) and (36).

Target state estimation

16. Calculate an estimate of the number of targets as $\hat{N}_{s,k} = \mathrm{round}\{W_{s,k}^{[I]}\}$.
17. Take the target state estimates $\hat{\mathbf{x}}_{s,k}^{(\nu)}$, $\nu = 1, \ldots, \hat{N}_{s,k}$ to be the means of the $\hat{N}_{s,k}$ GCs with the largest weights $\Omega_{s,k,\ell}^{[I]}$.

the local filter, etc.), which have to be performed before and/or after the dissemination/fusion iterations; and τ is the time duration of one dissemination/fusion iteration. With our algorithm, operations 3 and 4 contribute to t_{filt} and operations 8–10 and 13–15 contribute to t_{inter}. Here, t_{inter} is comparable to most other algorithms but t_{filt} is significantly smaller. In fact, for most other algorithms, t_{filt} is the total duration of all the filtering operations (cf. our operations 1–7), which includes also the computationally intensive likelihood

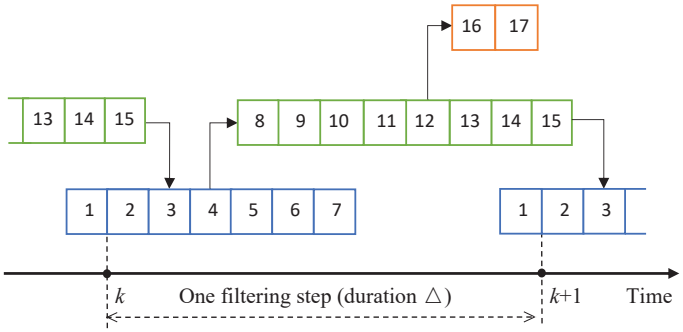

Figure 1: Parallelization of the operations of Algorithm 1. The numbers shown equal the operation numbers used in Algorithm 1. We note that the temporal duration of operations 11 and 12 is proportional to the number I of dissemination/fusion iterations, which is upper bounded by $I_{\max}$ in (37).

calculation (cf. operation 2) and, for, a particle-based implementation, also resampling (cf. operation 7). Thus, it follows from (37) that for our algorithm, $I_{\max}$ is significantly larger than for the other algorithms. This is an important advantage, as more dissemination/fusion iterations usually imply a better estimation accuracy.

9.2. Communication Cost

In one dissemination/fusion iteration of the proposed distributed PHD filter, each sensor s broadcasts to its neighbors a certain number of GC parameter sets, where each set consists of a weight, a d-dimensional mean vector, and a $d \times d$ symmetric covariance matrix. Thus, for each GC, $n_{\mathrm{GC}} \triangleq 1 + d + \frac{d(d+1)}{2}$ real values are broadcast by sensor s. In addition, sensor s broadcasts one cardinality estimate, which is a single real value. Let $n_{s,k}^{[i]}$ denote the number of GCs contained in the GM of sensor s in dissemination/fusion iteration i, before the fusion with the neighboring sensors is performed. Then the total number of real values broadcast by sensor s in one dissemination/fusion iteration is

$$N_{s,k}^{\mathrm{com[i]}} = n_{s,k}^{[i]} n_{\mathrm{GC}} + 1 = n_{s,k}^{[i]} \left(1 + d + \frac{d(d+1)}{2}\right) + 1. \tag{38}$$

Note that $N_{s,k}^{\mathrm{com[i]}}$ grows linearly with the number of GCs, $n_{s,k}^{[i]}$, and quadratically with the dimension of the target states, d, and it does not depend on the number of sensors, S. The last fact implies that the total communication cost for the entire network grows linearly with the network size S.

While expression (38) holds for both the GM flooding scheme of Section 6.1 and the GM average consensus scheme of Section 6.2, the communication costs of the two schemes are actually very different. In the case of the GM flooding scheme, the number of GCs broadcast is $n_{s,k}^{[i]} = \left|\mathcal{G}_{s,k}^{\mathrm{F}[i-1]}\right|$, which systematically grows with the iteration index i according to (19) or equivalently (22). In the case of the GM average consensus scheme, we have $n_{s,k}^{[i]} = \left|\mathcal{G}_{s,k}^{[i-1]}\right|$, which, according to Section 6.2, does not systematically grow with i because in each iteration a GC merging step is carried out. A quantitative characterization of $\left|\mathcal{G}_{s,k}^{[i-1]}\right|$ is difficult because the reduction of the number of GCs due to merging is situation-dependent—it is larger if the GCs are closer to each other.

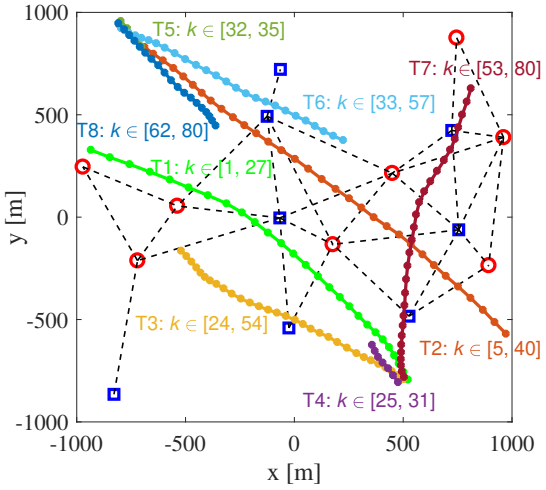

Figure 2: ROI, sensor network, and target trajectories. Blue squares and red circles indicate the positions of the linear and nonlinear sensors, respectively, black dashed lines indicate the communication links between neighboring sensors, and colored lines with dots indicate the target trajectories (with starting and ending times noted).

10. Simulation Study

10.1. Simulation Setup

10.1.1. Targets and Sensors

We simulated six targets that move in a square two-dimensional (2-D) region of interest (ROI) given by $[-1000\text{m}, 1000\text{m}] \times [-1000\text{m}, 1000\text{m}]$. The sensor network—consisting of 16 sensors—and the target trajectories are depicted in Fig. 2. The target states consist of 2-D position and 2-D velocity, i.e., $\mathbf{x}_k = [x_k \ \dot{x}_k \ y_k \ \dot{y}_k]^{\mathrm{T}}$. The target survival probability is $p_k^{\mathrm{S}}(\mathbf{x}_{k-1}) = 0.98$. The states of the surviving targets evolve independently according to a nearly constant velocity model, i.e., $\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{G}\mathbf{u}_k$, where $\mathbf{F} \in \mathbb{R}^{4\times4}$ and $\mathbf{G} \in \mathbb{R}^{4\times2}$ are as given in [56, Eq. (14)] with sampling period $\Delta = 1\,\text{s}$ and $\mathbf{u}_k$ is an independent and identically distributed (iid), zero-mean, Gaussian system process with standard deviation $5\,\text{m/s}^2$. The birth intensity function is $\gamma_k(\mathbf{x}_k) = 0.05 \cdot \mathcal{N}(\mathbf{x}_k; \mathbf{m}_1, \mathbf{Q}) + 0.05 \cdot \mathcal{N}(\mathbf{x}_k; \mathbf{m}_2, \mathbf{Q})$, where $\mathbf{m}_1 = [500\text{m} \ -20\text{m/s} \ -800\text{m} \ 30\text{m/s}]^{\mathrm{T}}$, $\mathbf{m}_2 = [-800\text{m} \ 30\text{m/s} \ 950\text{m} \ -30\text{m/s}]^{\mathrm{T}}$, and $\mathbf{Q} = \text{diag}\{400\text{m}^2, 100\text{m}^2/\text{s}^2, 400\text{m}^2, 100\text{m}^2/\text{s}^2\}$.

Eight of the 16 sensors acquire noisy position measurements with a fixed detection probability $p_{s,k}^{\mathrm{D}}(\mathbf{x}_k) = 0.9$. For these "linear" sensors, the measurement model is

$$\mathbf{z}_{s,k} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} v_{s,k}^{(1)} \\ v_{s,k}^{(2)} \end{bmatrix},$$

where $v_{s,k}^{(1)}$ and $v_{s,k}^{(2)}$ are iid zero-mean Gaussian with standard deviation 20m^2. The field of view of the linear sensors is the entire ROI. The other eight sensors are "nonlinear" sensors that acquire noisy range and bearing measurements with detection probability given by [57]

$$p_{s,k}^{\mathrm{D}}(\mathbf{x}_k) = 0.95 \cdot \frac{\mathcal{N}(\boldsymbol{\mu}_{\mathrm{D}}(\mathbf{x}_k); \mathbf{0}, 6000^2\mathbf{I}_2)}{\mathcal{N}(\mathbf{0}; \mathbf{0}, 6000^2\mathbf{I}_2)}.$$

Here, $\boldsymbol{\mu}_{\mathrm{D}}(\mathbf{x}_k) \triangleq [|x_k - x^{(s)}| \ |y_k - y^{(s)}|]^{\mathrm{T}}$, where $x^{(s)}$ and $y^{(s)}$ are the coordinates of sensor s. The range-bearing measurement model is

$$\mathbf{z}_{s,k} = \begin{bmatrix} \sqrt{(x_k - x^{(s)})^2 + (y_k - y^{(s)})^2} \\ \tan^{-1}\left(\frac{x_k - x^{(s)}}{y_k - y^{(s)}}\right) \end{bmatrix} + \begin{bmatrix} v_{s,k}^{(1)} \\ v_{s,k}^{(2)} \end{bmatrix},$$

where $v_{s,k}^{(1)}$ and $v_{s,k}^{(2)}$ are, individually, iid zero-mean Gaussian with standard deviation $\sigma_1 = 20\text{m}$ and $\sigma_2 = (\pi/90)\text{rad}$, respectively. The field of view of the nonlinear sensors is a disk of radius 3000m centered at the sensor position; this disk always covers the entire ROI. For both the linear and the nonlinear sensors, clutter is uniformly distributed over the sensor's field of view with an average number of r_c clutter measurements per time step, or equivalently clutter intensity $\kappa_{s,k}(\mathbf{z}_k) = r_c/2000^2$ for the linear sensors and $\kappa_{s,k}(\mathbf{z}_k) = r_c/(2\pi \cdot 3000)$ for the nonlinear sensors. We will consider the cases $r_c = 10$ and $r_c = 50$. The clutter measurements of different sensors are independent.

10.1.2. Local PHD Filters

We consider two scenarios. In the first scenario, all the local PHD filters use a particle-based implementation. In the second scenario, only the local PHD filters at the nonlinear sensor nodes use a particle-based implementation, whereas the local PHD filters at the linear sensor nodes use a GM-based implementation [3, 10]. The results for the second scenario demonstrate the applicability of our distributed PHD filter in heterogeneous networks combining particle-based and GM-based local PHD filters.

We compare the performance and computing time of the following particle-based PHD filters:

- The proposed distributed PHD filter, which will be briefly referred to as *AA-F-IS* or *AA-C-IS* depending on whether flooding (F) or average consensus (C) is used as the dissemination/fusion scheme.

- A modified version of the GA fusion-based, particle-based, distributed PHD filter proposed in [12], briefly referred to as *GA-EMD*. In [12], two important steps are (i) a conversion of the particle representation of the PHD into a kernel-based representation, and (ii) the construction of the multitarget exponential mixture density (EMD). Regarding the first step, we replaced the clustering algorithm for kernel function learning proposed in [12]—which we observed in our simulations to be computationally intensive and potentially unstable—with our particles-GM conversion algorithm from Section 5. Regarding the second step, we use our IS method for GM-particles conversion (see Section 7) for updating the fused particles. Finally, we do not employ the sophisticated strategy for online adjustment of the fusion weights proposed in [12] but use fixed Metropolis weight (see Section 6.2), which have been widely used for GA-based GM-PHD fusion [4].

  The resulting modification of the EMD fusion method of [12] is more computationally efficient, although—as shown later—it is still considerably less efficient than our proposed fusion method. Moreover, just as the filter of [12], it has a significantly higher communication cost because it communicates both the particles and the kernel/GM parameters. For this reason, using flooding for dissemination/fusion is infeasible, and hence we only use the average consensus scheme.

- A modified version of our proposed distributed PHD filter, in which the GM–particles conversion is done via the standard sampling (SS) method—i.e., sampling directly from the fused GM-PHD $D_{s,k}^{\text{GM}[I]}(\mathbf{x})$—instead of our IS method from Section 7. We consider this filter to compare the IS method with the SS method. We refer to it as *AA-F-SS* or *AA-C-SS* depending on the dissemination/fusion scheme employed.

- A noncooperative PHD filter in which each local PHD filter relies solely on its own local measurements and does not communicate with other local PHD filters.

- A centralized multisensor PHD filter based on the iterated-corrector (IC) principle [58]. This is a commonly used approximation of the (computationally intractable) optimal multisensor PHD filter [58]. The centralized filter has direct access to the measurements of all the sensors. The measurements of different sensors are applied in an arbitrary order to iteratively update the PHD; more specifically, the PHD updated by using the measurements of one sensor are interpreted as a prior PHD and updated by using the measurements of the next sensor, etc. Since a GM representation is no longer calculated, the target state estimates are extracted from the PHD by means of the k-means clustering method [2].

The local PHD filters use systematic resampling [55], and they adjust the number of particles via resampling to be $200 \cdot \hat{N}_{s,k}^{\text{local}}$ if $\hat{N}_{s,k}^{\text{local}} \geq 0.5$ and 100 otherwise, where $\hat{N}_{s,k}^{\text{local}} = \text{round}\{W_{s,k}\}$. (Here, we use $W_{s,k}$ and not $W_{s,k}^{[I]}$ because in the resampling step, $W_{s,k}^{[I]}$ is not available yet.) The target state estimates $\hat{\mathbf{x}}_k^{(\nu)}$ are calculated as described in Section 8.2 (except in the centralized filter as explained above). The threshold defining $\Omega_{s,k}(\mathbf{z})$ (see Section 5) is $T_\Omega = 0.3$. The consensus-based filters (AA-C-SS/IS and GA-EMD) perform GC merging in each consensus iteration (see Section 6.2); GCs are merged if their Mahalanobis distance is smaller than 2 [53].

For each of the two scenarios, we performed 100 simulation runs using the target trajectories shown in Fig. 2 and randomly generated measurement noise and initial particles. Each simulation run consists of 80 time steps.

10.2. First Scenario—Particle-based Local PHD Filters

In the first scenario, all the local PHD filters use a particle-based implementation.

10.2.1. Tracking Accuracy

We quantify the target detection and position estimation performance of the filters by the mean optimal subpattern assignment (OSPA) error [59] with cutoff parameter $c = 1000\,\text{m}$ and order $p = 2$. More specifically, we consider the average of the OSPA errors obtained by all the sensors, referred to as *network OSPA error* (briefely N-OSPA) and the average of the network OSPA errors over all the 80 time steps, referred to as *time-averaged network OSPA error* (TN-OSPA).

We first consider the case $r_c = 10$. Fig. 3(a) shows the N-OSPA of the distributed PHD filters using $I = 5$ dissemination/fusion iterations, as well as of the noncooperative PHD filter and the centralized filter, versus time k. One can see that the D-PHD filters almost always have a significantly smaller N-OSPA than the centralized filter, which, in turn, almost always has a significantly smaller N-OSPA than the noncooperative PHD filter.

Fig. 3(b) shows the TN-OSPA versus the number I of dissemination/fusion iterations. For growing I, the TN-OSPA of all fusion-based filters decreases quite fast initially but for larger values of I it decreases more slowly (in the case of the consensus-based filters) or it stays roughly constant (in the case of the flooding-based filters) or it even starts increasing again (in the case of GA-EMD). Regarding the flooding-based filters, we recall from Section 6.1 that the flooding dissemination of the GM parameters is already complete when I equals the network diameter $R = 5$, and thus no further gains can be achieved for $I \geq 6$.

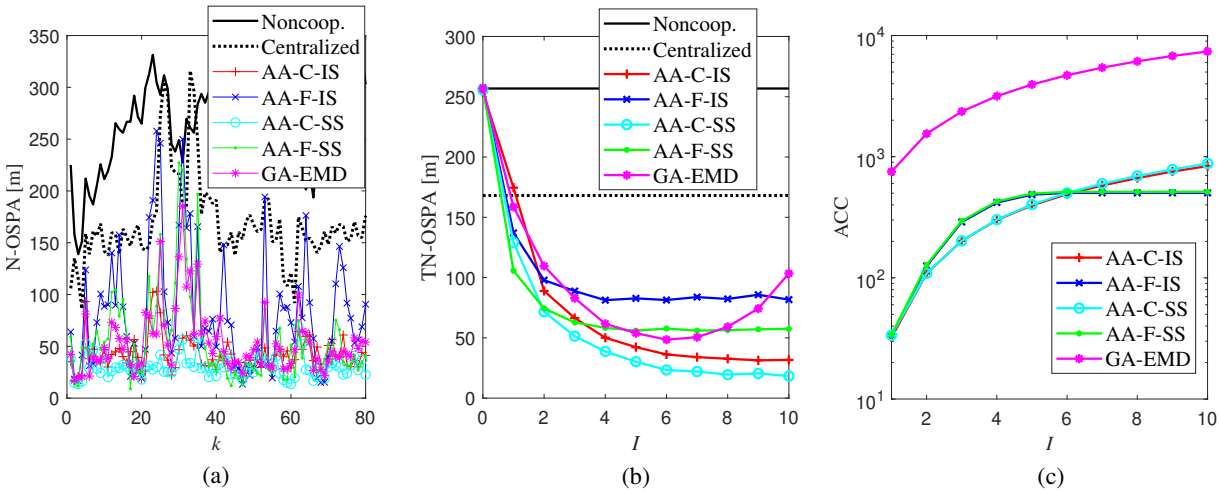

Figure 3: Results for the first scenario with $r_c = 10$: (a) Network OSPA error versus time k (here, the distributed filters use $I = 5$ dissemination/fusion iterations). (b) Time-averaged network OSPA error versus number of dissemination/fusion iterations I. (c) Average communication cost versus I.

The fusion-based filters using $I \geq 2$ dissemination/fusion iterations significantly outperform the centralized filter. Indeed, it is well known that the performance of the IC approach to multisensor PHD filtering is sensitive to the sensor order and usually only moderate. Regarding the increase of the TN-OSPA of GA-EMD for $I \geq 7$, we conjecture that it is due to the fact that a missed detection at any single sensor can degrade the performance of GA fusion significantly, and the probability of such a missed detection increases when more sensors are involved. We note that a similar increase of the OSPA for additional GA dissemination/fusion iterations was reported in [8] (see especially [8, Fig. 2(b)]) and in [22] (see especially [22, Figs. 3 and 4]). In particular, it was demonstrated in [22] that in the case of low detection probability, the performance of GA fusion deteriorates when the number of consensus steps is increased, and that GA fusion may even perform worse than no fusion. It is furthermore seen in Fig. 3(b) that the TN-OSPA of GA-EMD is larger than that of AA-C-IS/SS (except for $I = 1$, where according to Fig. 3(b) it is slightly smaller than that of AA-C-IS).

The OSPA performance of the SS-based filters (AA-F-SS and AA-C-SS) is seen to be better than that of the corresponding IS-based filters (AA-F-IS and AA-C-IS, respectively). This is because sampling directly from $D_{s,k}^{\mathrm{GM}[I]}(\mathbf{x})$ represents $D_{s,k}^{\mathrm{GM}[I]}(\mathbf{x})$ more accurately than the indirect sampling performed by our IS method. (However, we recall that the IS method enables the far-reaching parallelization of filtering and fusion operations described in Section 9.1.) Finally, the consensus-based filters (AA-C-SS and AA-C-IS) outperform the flooding-based filters (AA-F-SS and AA-F-IS, respectively); the only exception is $I = 1$, where the consensus and flooding schemes differ merely by the choice of the fusion weights (uniform and Metropolis weights, respectively). This superiority of the GM consensus scheme (for $I \geq 2$) is unexpected, since flooding yields a faster dissemination of the GM parameters than consensus. A possible reason is the GC merging performed by the GM consensus scheme in each fusion iteration. In this context, an interesting observation is that GA-EMD—which is also consensus-based and performs GC merging—outperforms AA-F-IS for $3 \leq I \leq 9$. In additional simulations for various scenarios, we observed that the performance of consensus-based PHD filter algorithms with GC merging, including AA-C-SS/IS and GA-EMD, is highly sensitive to the threshold of the Mahalanobis distance used for GC merging: we found that threshold 2 yields the best filter performance, whereas other thresholds can lead to a significantly poorer performance.

Filter	Average Computing Time [s]
Noncooperative	0.079
Centralized	0.770
AA-F-SS	0.181
AA-C-SS	0.347
AA-C-IS	0.387
AA-F-IS	0.558
GA-EMD	1.837

Table 1: Results for the first scenario with $r_c = 10$: Average computing time of one filtering step. The distributed filters use $I = 5$ dissemination/fusion iterations.

10.2.2. Communication Cost

We measure the *average communication cost* (ACC) of the various distributed (fusion-based) filters by the number of real values broadcast by a sensor to its neighbors during all the dissemination/fusion iterations performed at one time step, averaged over all the sensors, time steps, and simulation runs. Note that in addition to one real value for the cardinality estimate, only GC parameters are broadcast in AA-F/C-IS and AA-F/C-SS whereas in GA-EMD, both GC parameters and unweighted particles (i.e., the particles after the resampling step) are broadcast. Here, each unweighted particle amounts to four real values.

Fig. 3(c) shows the ACC versus I. The increase of the ACC of GA-EMD and AA-C-IS/SS with I is an expected result because the ACC was defined as the average total communication cost for all the I dissemination/fusion iterations. The ACC of AA-F-IS/SS increases up to $I = 5$ but stays constant afterwards. This is also expected because, as mentioned earlier, the flooding dissemination is already complete when $I = R = 5$, and thus no additional information needs to be communicated for $I \geq 6$. The ACC of GA-EMD is seen to be larger by about one order of magnitude than that of the other filters; this is because GA-EMD communicates a large number of particles in addition to GC parameters. Furthermore, the ACC of the flooding-based filters is larger than that of the consensus-based filters for I between 2 and 5, and smaller for $I \geq 7$. At this point, we recall from Section 9.2 that the communication cost of the consensus-based filters strongly depends on the GC merging. Using a larger threshold for the Mahalanobis distance (so that more GCs are merged) would result in a smaller communication cost but also in a poorer tracking accuracy. Finally, AA-F-SS and AA-F-IS are seen to have almost the same ACC, and similarly for AA-C-SS and AA-C-IS. This is because the choice of the GM–particles conversion method—SS or IS—has only little effect on the communication cost.

10.2.3. Computational Complexity

Finally, we quantify the computational complexity of the various filters by the *average computing time* of each filtering step (corresponding to each time step $k \to k+1$) in one local filter, where the averaging is over all the local PHD filters, time steps, and simulation runs. The computing times were obtained using MATLAB implementations on an Intel Core M-5Y71 CPU. Table 1 shows the average computing time for the distributed PHD filters using $I = 5$ dissemination/fusion iterations, for the noncooperative filter, and for the centralized filter. It is seen that GA-EMD is significantly more complex than the other distributed filters. Furthermore, AA-F-IS and AA-C-IS are more complex than AA-F-SS and AA-C-SS; this is because the IS method is more complex than the SS method. AA-F-IS is more complex than AA-C-IS, due to the larger

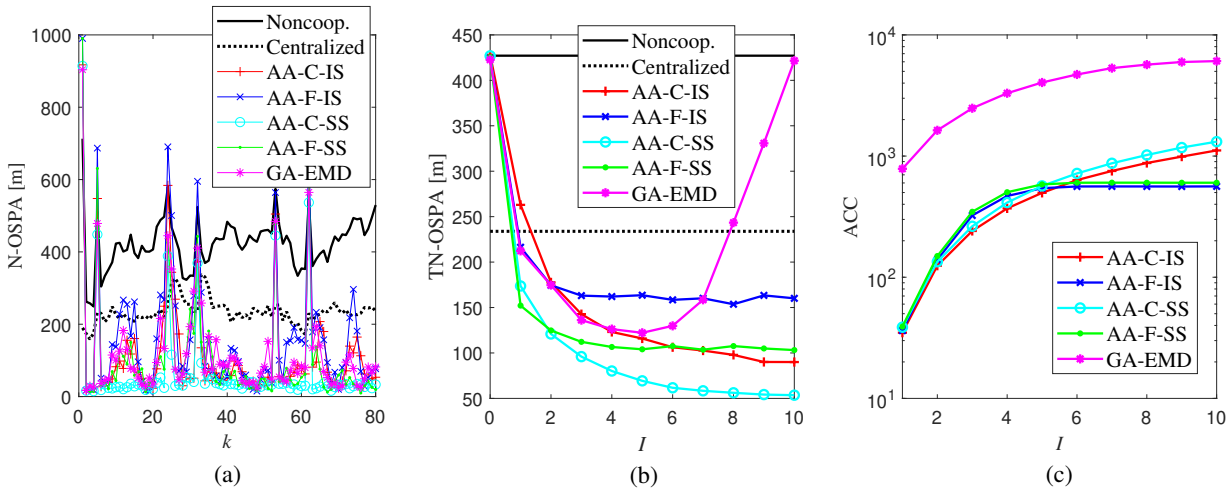

Figure 4: Results for the first scenario with $r_c = 50$: (a) Network OSPA error versus time k (here, the distributed filters use $I = 5$ dissemination/fusion iterations). (b) Time-averaged network OSPA error versus number of dissemination/fusion iterations I. (c) Average communication cost versus I.

number of GCs that are processed. Indeed, in AA-C-IS, the number of GCs is reduced by GC merging, and the complexity of the GM merging operations is considerably smaller than the added complexity of AA-F-IS caused by the additional GCs. On the other hand, AA-F-SS is less complex than AA-C-SS. Here, the reason is that the SS method employed by AA-F-SS and AA-C-SS has a low complexity, and thus the complexity of the merging operations performed by AA-C-SS is larger than the added complexity of AA-F-SS caused by the additional GCs. Finally, the centralized filter is more complex than the AA fusion-based filters but still less complex than GA-EMD.

10.2.4. Heavy Clutter

We also tested the performance of the different filters using a much higher clutter rate, namely $r_c = 50$, corresponding to clutter intensity $\kappa_{s,k}(\mathbf{z}_k) = 50/2000^2 = 1.25 \cdot 10^{-5}$ for the linear sensors and $\kappa_{s,k}(\mathbf{z}_k) = 50/(2\pi \cdot 3000) \approx 2.65 \cdot 10^{-3}$ for the nonlinear sensors. All the other parameters and filter settings were as before. The OSPA results are shown in Fig. 4; as expected, they are worse than those for $r_c = 10$ (cf. Fig. 3). However, the relative OSPA performance of the different filters is similar to that for $r_c = 10$ except for two differences. First, the GA-EMD filter performs better—in the sense of a lower TN-OSPA—than the AA-C-IS filter for $I \geq 3$ (whereas in the case of $r_c = 10$, a better performance was observed only for $I = 1$). Second, the performance of the GA-EMD filter degrades more strongly (compared to the case $r_c = 10$) for increasing $I \geq 5$; note that, by contrast, all the AA fusion-based filters always benefit from an increasing I. These results confirm that our proposed AA fusion scheme remains effective even in heavily cluttered environments, and they also demonstrate a weakness of GA fusion in such environments [60].

The ACCs of the various distributed filters, shown in Fig. 4(c), are similar to those obtained for $r_c = 10$. Thus, the stronger clutter does not significantly increase the communication requirements. The underlying reason is that the communication requirements depend on the number of significant GCs and the clutter is unlikely to generate significant GCs. This demonstrates that our proposed AA fusion scheme is able to reject GCs that correspond to false alarms.

Finally, Table 2 shows that the average computing times for the various filters are significantly higher than those obtained for $r_c = 10$ (cf. Table 1). This is not surprising, as the number of measurements increases

Filter	Average Computing Time [s]
Noncooperative	0.158
Centralized	1.735
AA-F-SS	0.278
AA-C-SS	0.583
AA-C-IS	0.592
AA-F-IS	0.706
GA-EMD	2.462

Table 2: Results for the first scenario with $r_c = 50$: Average computing time of one filtering step. The distributed filters use $I = 5$ dissemination/fusion iterations.

approximately linearly with r_c and the computational complexity of the PHD filter increases linearly with the number of measurements.

10.3. Second Scenario—Particle-based and GM-based Local PHD Filters

Next, we study a heterogeneous network where the eight nonlinear sensor nodes use a particle-based local PHD filter and the eight linear sensor nodes use a GM-based local PHD filter [3, 10] (briefly referred to as GM-PHD filter). The sensor network topology and the target trajectories are as before (see Fig. 2). The clutter rate is $r_c = 10$. The GM-PHD filters use at most 100 GCs. For mixture reduction, following [3], they remove GCs with a weight smaller than 10^{-4} and merge GCs with a Mahalanobis distance smaller than 4. (We note that here, the Mahalanobis distance threshold 4 performed better than the threshold 2 that we used in the consensus-based particle-PHD filters in Section 10.2.) Furthermore, for fusing their local GM with the GMs of the other sensors, the GM-PHD filters perform a straightforward union of the GM parameter sets and subsequently adjust the weights using the cardinality averaging method discussed in Section 8.1. The combination—within the sensor network—of the GM-PHD filters with the particle-based AA-F/C-SS/IS filters will be briefly referred to as "AA-F/C-SS/IS." We no longer consider GA-EMD as it cannot be combined with a GM-PHD filter in a straightforward fashion (i.e., without conversions between particle and GM representations). Also, we do not consider a centralized PHD filter because we would have to choose between a GM implementation and an SMC implementation, either of which would not provide a meaningful comparison with our hybrid GM/SMC distributed PHD filter.

The simulation results for this scenario, shown in Fig. 5 and Table 3, are generally similar to those for the first scenario (see Fig. 3 and Table 1). A difference is that now AA-F-IS and AA-F-SS have a smaller TN-OSPA than, respectively, AA-C-IS and AA-C-SS for $I \leq 4$, instead of only for $I = 1$ (as was the case in the first scenario). This is because now half of the local filters are GM-PHD filters, for which flooding-based fusion performs better than consensus-based fusion [10].

11. Conclusion

We proposed a distributed PHD (D-PHD) filter where the local filters use a particle-based implementation to support nonlinear/non-Gaussian system models, but the fusion of the local PHDs is based on a Gaussian mixture (GM) representation to reduce communication and enable an easy combination with GM-based local filters. Our D-PHD filter differs from most existing filters in that it seeks to compute the arithmetic average (AA) of the local PHDs, rather than the geometric average (GA). Two noteworthy components of

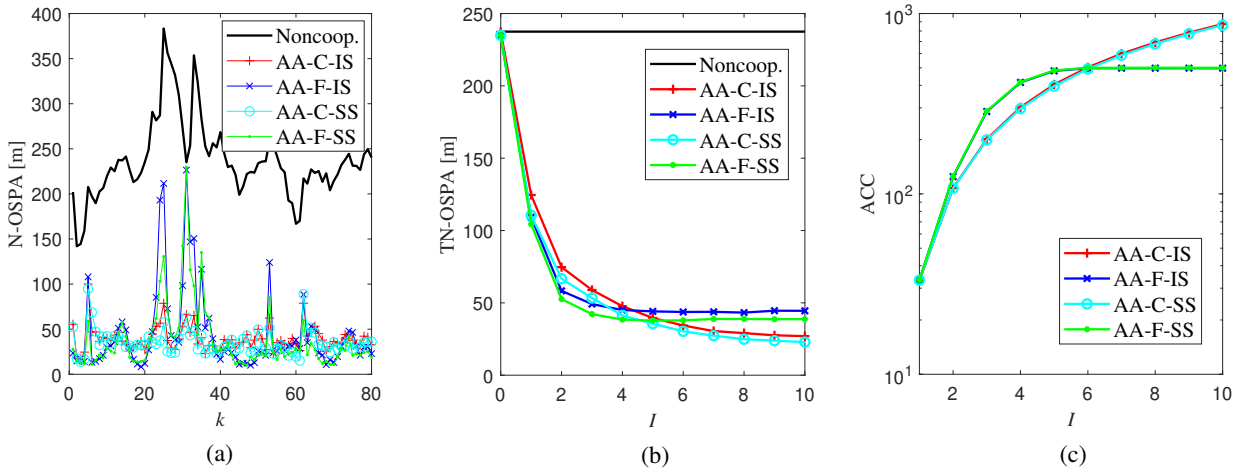

Figure 5: Results for the second scenario with $r_c = 10$: (a) Network OSPA error versus time k (here, the distributed filters use $I = 5$ dissemination/fusion iterations). (b) Time-averaged network OSPA error versus number I of dissemination/fusion iterations. (c) Average communication cost versus I.

Filter	Average Computing Time [s]
Noncooperative	0.096
AA-F-SS	0.233
AA-C-SS	0.353
AA-C-IS	0.381
AA-F-IS	0.422

Table 3: Results for the second scenario with $r_c = 10$: Average computing time of one filtering step. The distributed filters use $I = 5$ dissemination/fusion iterations.

our D-PHD filter algorithm are (i) a "significance-based" method for converting particle representations into GM representations, which reduces communication and complexity, and (ii) an importance sampling method for converting the fused GMs into particle representations, which enables a parallelization of filtering and fusion operations. This parallelization is especially advantageous when the sensing rate is high and/or the duration of one dissemination/fusion iteration is large.

An experimental comparison of our filter with a state-of-the-art filter using GA fusion showed that, in the considered scenarios, consensus-based AA fusion outperforms consensus-based GA fusion in terms of estimation accuracy, complexity, and communication cost. Our simulations also showed that consensus-based AA fusion can outperform flooding-based AA fusion in terms of both estimation accuracy and communication cost. We expect that this advantage of AA fusion can be further increased by using more sophisticated mixture reduction schemes such as [61, 62]. Further interesting directions of future research include the use of our D-PHD filter in networks with time-varying topology and for the problem of joint cooperative sensor localization and multitarget tracking.

Acknowledgments

This work was partially supported by the Marie Skłodowska-Curie Individual Fellowship (H2020-MSCA-IF-2015) under Grant 709267 and by the Austrian Science Fund (FWF) under Grant P32055-N31.

Appendix: Proof of Eq. (32)

To show that the choice of $P_{j'}$ in (32) yields (31), we note that $N_{s,k}^{(j')}$ can be written as $N_{s,k}^{(j')} = \sum_{j=1}^{\tilde{J}_{s,k}} I[\tilde{\mathbf{x}}_{s,k}^{(j)} = \mathbf{x}_{s,k}^{(j')}]$, where $I[\tilde{\mathbf{x}}_{s,k}^{(j)} = \mathbf{x}_{s,k}^{(j')}]$ equals 1 if $\tilde{\mathbf{x}}_{s,k}^{(j)} = \mathbf{x}_{s,k}^{(j')}$ and 0 otherwise. Thus,

$$\mathrm{E}[N_{s,k}^{(j')}|\xi_{s,k}] = \sum_{j=1}^{\tilde{J}_{s,k}} \mathrm{E}[I[\tilde{\mathbf{x}}_{s,k}^{(j)} = \mathbf{x}_{s,k}^{(j')}]|\xi_{s,k}]. \tag{39}$$

Now

$$\mathrm{E}[I[\tilde{\mathbf{x}}_{s,k}^{(j)} = \mathbf{x}_{s,k}^{(j')}] = 1 \cdot \Pr[I[\tilde{\mathbf{x}}_{s,k}^{(j)} = \mathbf{x}_{s,k}^{(j')}]|\xi_{s,k}] + 0 \cdot \Pr[I[\tilde{\mathbf{x}}_{s,k}^{(j)} = \mathbf{x}_{s,k}^{(j')}]|\xi_{s,k}] = P_{j'},$$

whence (39) becomes

$$\mathrm{E}[N_{s,k}^{(j')}|\xi_{s,k}] = \sum_{j=1}^{\tilde{J}_{s,k}} P_{j'} = \tilde{J}_{s,k} P_{j'} = \mathrm{round}\{N_{\mathrm{p}} W_{s,k}\} P_{j'} \approx N_{\mathrm{p}} W_{s,k} P_{j'} = N_{\mathrm{p}} w_{s,k}^{(j')},$$

where (30) and (32) have been used. Hence, to within a rounding error (caused by replacing $\mathrm{round}\{N_{\mathrm{p}} W_{s,k}\}$ with $N_{\mathrm{p}} W_{s,k}$), $\mathrm{E}[N_{s,k}^{(j')}|\xi_{s,k}]$ equals $N_{\mathrm{p}} w_{s,k}^{(j')}$, as postulated in (31).

References

[1] R. P. S. Mahler, Multitarget Bayes filtering via first-order multitarget moments, IEEE Trans. Aerosp. Electron. Syst. 39 (2003) 1152–1178.

[2] B.-N. Vo, S. Singh, A. Doucet, Sequential Monte Carlo methods for multitarget filtering with random finite sets, IEEE Trans. Aerosp. Electron. Syst. 41 (2005) 1224–1245.

[3] B.-N. Vo, W. K. Ma, The Gaussian mixture probability hypothesis density filter, IEEE Trans. Signal Process. 54 (2006) 4091–4104.

[4] G. Battistelli, L. Chisci, C. Fantacci, A. Farina, A. Graziano, Consensus CPHD filter for distributed multitarget tracking, IEEE J. Sel. Topics Signal Process 7 (2013) 508–520.

[5] G. Battistelli, L. Chisci, C. Fantacci, A. Farina, R. P. S. Mahler, Distributed fusion of multitarget densities and consensus PHD/CPHD filters, in: Proc. SPIE, volume 9474, pp. 94740E–94740E–15.

[6] M. Gunay, U. Orguner, M. Demirekler, Chernoff fusion of Gaussian mixtures based on sigma-point approximation, IEEE Trans. Aerosp. Electron. Syst. 52 (2016) 2732–2746.

[7] W. Yi, M. Jiang, S. Li, B. Wang, Distributed sensor fusion for RFS density with consideration of limited sensing ability, in: Proc. FUSION 2017, Xi'an, China, pp. 1610–1615.

[8] J. Y. Yu, M. Coates, M. Rabbat, Distributed multi-sensor CPHD filter using pairwise gossiping, in: Proc. IEEE ICASSP 2016, Shanghai, China, pp. 3176–3180.

[9] T. Li, J. Corchado, S. Sun, On generalized covariance intersection for distributed PHD filtering and a simple but better alternative, in: Proc. FUSION 2017, Xi'an, China, pp. 808–815.

[10] T. Li, J. Corchado, S. Sun, Partial consensus and conservative fusion of Gaussian mixtures for distributed PHD fusion, IEEE Trans. Aerosp. Electron. Syst. 55 (2019) 2150–2163.

[11] M. Üney, S. Julier, D. Clark, B. Ristić, Monte Carlo realisation of a distributed multi-object fusion algorithm, in: Proc. SSPD 2010, London, UK.

[12] M. Üney, D. E. Clark, S. J. Julier, Distributed fusion of PHD filters via exponential mixture densities, IEEE J. Sel. Topics Signal Process 7 (2013) 521–531.

[13] A. K. Gostar, R. Hoseinnezhad, A. Bab-Hadiashar, Cauchy-Schwarz divergence-based distributed fusion with Poisson random finite sets, in: Proc. ICCAIS 2017, Chiang Mai, Thailand, pp. 112–116.

[14] T. Li, V. Elvira, H. Fan, J. M. Corchado, Local diffusion-based distributed SMC-PHD filtering using sensors with limited sensing range, IEEE Sensors J. 19 (2019) 1580–1589.

[15] J. K. Uhlmann, Dynamic map building and localization: New theoretical foundations, Ph.D. thesis, University of Oxford, UK, 1995.

[16] S. Julier, J. Uhlmann, General decentralized data fusion with covariance intersection (CI), in: D. Hall, J. Llinas (Eds.), Handbook of Data Fusion, CRC Press, Boca Raton, FL, USA, 2001, pp. 1–25.

[17] T. Bailey, S. Julier, G. Agamennoni, On conservative fusion of information with unknown non-Gaussian dependence, in: Proc. FUSION 2012, Singapore, pp. 1876–1883.

[18] R. P. S. Mahler, Optimal/robust distributed data fusion: A unified approach, in: Proc. SPIE, volume 4052, pp. 128–138.

[19] D. Clark, S. Julier, R. Mahler, B. Ristic, Robust multi-object sensor fusion with unknown correlations, in: Proc. SSPD 2010, London, UK.

[20] R. P. S. Mahler, Toward a theoretical foundation for distributed fusion, in: D. Hall, C.-Y. Chong, J. Llinas, M. Liggins (Eds.), Distributed Data Fusion for Network-Centric Operations, CRC Press, Boca Raton, FL, USA, 2012, pp. 199–224.

[21] A. E. Abbas, A Kullback-Leibler view of linear and log-linear pools, Decision Analysis 6 (2009) 25–37.

[22] L. Gao, G. Battistelli, L. Chisci, Multiobject fusion with minimum information loss, IEEE Signal Process. Lett. (2019).

[23] G. Battistelli, L. Chisci, C. Fantacci, N. Forti, A. Farina, A. Graziano, Distributed peer-to-peer multitarget tracking with association-based track fusion, in: Proc. FUSION 2014, Salamanca, Spain, pp. 1–7.

[24] M. Üney, J. Houssineau, E. Delande, S. J. Julier, D. E. Clark, Fusion of finite set distributions: Pointwise consistency and global cardinality, IEEE Trans. Aerosp. Electron. Syst. (2019). To be published.

[25] R. L. Streit, Multisensor multitarget intensity filter, in: Proc. FUSION 2008, Cologne, Germany.

[26] H. Kim, K. Granström, , L. Gao, G. Battistelli, S. Kim, H. Wymeersch, 5g mmWave cooperative positioning and mapping using multi-model PHD filter and map fusion (2019). [Online] arXiv:1908.09806.

[27] K. Da, T. Li, Y. Zhu, H. Fan, Q. Fu, Kullback-Leibler averaging for multitarget density fusion, in: Proc. DCAI 2019, Avila, Spain, pp. 253–261.

[28] M. R. Fréchet, Les éléments aléatoires de nature quelconque dans un espace distancié, Annales de l'institut Henri Poincaré 10 (1948) 215–310.

[29] T. Li, X. Wang, Y. Liang, Q. Pan, On arithmetic average fusion and its application for distributed multi-Bernoulli multitarget tracking, techRxiv.org: 10.36227/techrxiv.11599902 (2020).

[30] R. P. S. Mahler, The multisensor PHD filter: II. Erroneous solution via Poisson magic, in: Proc. SPIE 2009, volume 7336, pp. 7336–12.

[31] S. Nagappa, D. E. Clark, R. Mahler, Incorporating track uncertainty into the OSPA metric, in: Proc. FUSION 2011, Chicago, IL, USA.

[32] T. Li, H. Fan, J. García, J. M. Corchado, Second-order statistics analysis and comparison between arithmetic and geometric average fusion: Application to multi-sensor target tracking, Inf. Fusion 51 (2019) 233–243.

[33] K. Da, T. Li, Y. Zhu, Q. Fu, A computationally efficient approach for distributed sensor localization and multitarget tracking, IEEE Commun. Lett. 24 (2019) 335–338.

[34] N. Lehrer, O. Tslil, A. Carmi, Log-linear Chernoff fusion for distributed particle filtering, in: Proc. FUSION 2019, Ottawa, Canada.

[35] M. Üney, D. E. Clark, S. J. Julier, Information measures in distributed multitarget tracking, in: Proc. Fusion 2011, Chicago, IL, USA.

[36] L. Zhao, P. Ma, X. Su, H. Zhang, A new multi-target state estimation algorithm for PHD particle filter, in: Proc. FUSION 2010, Edinburgh, Scotland, UK.

[37] B. Ristic, D. Clark, B.-N. Vo, Improved SMC implementation of the PHD filter, in: Proc. FUSION 2010, Edinburgh, Scotland, UK.

[38] M. Schikora, W. Koch, R. Streit, D. Cremers, Sequential Monte Carlo method for multi-target tracking with the intensity filter, in: P. Georgieva, L. Mihaylova, L. C. Jain (Eds.), Advances in Intelligent Signal Processing and Data Mining: Theory and Applications, Springer, Heidelberg, Germany, 2012, pp. 55–87.

[39] C. P. Robert, G. Casella, Monte Carlo Statistical Methods, Springer, Secaucus, NJ, USA, 2005.

[40] R. P. S. Mahler, Statistical Multisource-Multitarget Information Fusion, Artech House, Norwood, MA, USA, 2007.

[41] S. S. Singh, B.-N. Vo, A. Baddeley, S. Zuyev, Filters for spatial point processes, SIAM J. Contr. Opt. 48 (2009) 2275–2295.

[42] S. M. Kay, Fundamentals of Statistical Signal Processing: Estimation Theory, Prentice-Hall, Upper Saddle River, NJ, USA, 1993.

[43] T. Heskes, Selecting weighting factors in logarithmic opinion pools, in: M. Kearns, S. Solla, D. Cohn (Eds.), Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, USA, 1998, pp. 266–272.

[44] T. Li, F. Hlawatsch, P. M. Djuríc, Cardinality-consensus-based PHD filtering for distributed multitarget tracking, IEEE Signal Process. Lett. 26 (2019) 49–53.

[45] M. Coates, Distributed particle filters for sensor networks, in: Proc. IPSN 2004, New York, NY, USA, pp. 99–107.

[46] X. Sheng, Y. H. Hu, P. Ramanathan, Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network, in: Proc. IPSN 2005, Los Angeles, CA, USA, pp. 181–188.

[47] D. Gu, Distributed particle filter for target tracking, in: Proc. IEEE ICRA 2007, Rome, Italy, pp. 3856–3861.

[48] O. Hlinka, F. Hlawatsch, P. M. Djuric, Distributed particle filtering in agent networks: A survey, classification, and comparison, IEEE Signal Process. Mag. 30 (2013) 61–81.

[49] T. Li, J. Corchado, J. Prieto, Convergence of distributed flooding and its application for distributed Bayesian filtering, IEEE Trans. Signal Inf. Process. Netw. 3 (2017) 580–591.

[50] J. Li, A. Nehorai, Distributed particle filtering via optimal fusion of Gaussian mixtures, IEEE Trans. Signal Inf. Process. Netw. 4 (2018) 280–292.

[51] R. Olfati-Saber, J. A. Fax, R. M. Murray, Consensus and cooperation in networked multi-agent systems, Proc. IEEE 95 (2007) 215–233.

[52] L. Xiao, S. Boyd, Fast linear iterations for distributed averaging, Syst. Control. Lett. 53 (2004) 65–78.

[53] D. J. Salmond, Mixture reduction algorithms for target tracking in clutter, in: Proc. SPIE, volume 1305, pp. 434–445.

[54] S. Reece, S. Roberts, Generalised covariance union: A unified approach to hypothesis merging in tracking, IEEE Trans. Aerosp. Electron. Syst. 46 (2010) 207–221.

[55] T. Li, M. Bolić, P. M. Djurić, Resampling methods for particle filtering: Classification, implementation, and strategies, IEEE Signal Process. Mag. 32 (2015) 70–86.

[56] X. R. Li, V. P. Jilkov, Survey of maneuvering target tracking. Part I. Dynamic models, IEEE Trans. Aerosp. Electron. Syst. 39 (2003) 1333–1364.

[57] B.-T. Vo, B.-N. Vo, A. Cantoni, The cardinality balanced multi-target multi-Bernoulli filter and its implementations, IEEE Trans. Signal Process. 57 (2009) 409–423.

[58] R. Mahler, The multisensor PHD filter: I. General solution via multitarget calculus, in: Proc. SPIE, volume 7336.

[59] D. Schuhmacher, B.-T. Vo, B.-N. Vo, A consistent metric for performance evaluation of multi-object filters, IEEE Trans. Signal Process. 56 (2008) 3447–3457.

[60] B. Wang, W. Yi, R. Hoseinnezhad, S. Li, L. Kong, X. Yang, Distributed fusion with multi-Bernoulli filter based on generalized covariance intersection, IEEE Trans. Signal Process. 65 (2017) 242–255.

[61] D. F. Crouse, P. Willett, K. Pattipati, L. Svensson, A look at Gaussian mixture reduction algorithms, in: Proc. FUSION 2011, Chicago, IL, USA.

[62] T. Ardeshiri, K. Granström, E. Ozkan, U. Orguner, Greedy reduction algorithms for mixtures of exponential family, IEEE Signal Process. Lett. 22 (2015) 676–680.