

Article

Distributed Space Debris Tracking with Consensus Labeled Random Finite Set Filtering [†]

Baishen Wei ^{1,*} and Brett Nener ² ¹ School of Mechanical and Electric Engineering, Guangzhou University, Guangzhou 510006, China² School of Electrical, Electronic, and Computer Engineering, the University of Western Australia, Crawley, WA 6009, Australia; brett.nener@uwa.edu.com

* Correspondence: baishen@gzhu.edu.cn; Tel.: +86-180-2724-7271

[†] This paper is an extended version of the conference paper: Baishen, W.; Brett, N. Consensus labeled multi-Bernoulli filtering for distributed space debris tracking. In Proceedings of the 2017 International Conference on Control, Automation and Information Sciences (ICCAIS), Chiang Mai, Thailand, 31 October–1 November 2017.

Received: 25 July 2018; Accepted: 5 September 2018; Published: 7 September 2018



Abstract: Space debris tracking is a challenge for spacecraft operation because of the increasing number of both satellites and the amount of space debris. This paper investigates space debris tracking using marginalized δ -generalized labeled multi-Bernoulli filtering on a network of nodes consisting of a collection of sensors with different observation volumes. A consensus algorithm is used to achieve the global average by iterative regional averages. The sensor network can have unknown or time-varying topology. The proposed space debris tracking algorithm provides an efficient solution to the key challenges (e.g., *detection uncertainty*, *data association uncertainty*, *clutter*, etc.) for space situational awareness. The performance of the proposed algorithm is verified by simulation results.

Keywords: space debris; marginalized δ -GLMB; distributed; consensus

1. Introduction

This paper presents a distributed space objects tracking approach in the context of space situational awareness (SSA) [1]. SSA is the ability to track and predict the velocity and location of space objects in orbit around the Earth. SSA has become a significant concern for both commercial and military systems. A U.S. Iridium 33 communications satellite was struck by a defunct Russian Cosmos 2251 military communications satellite in February 2009. Collisions like this generate space debris, which then increases the likelihood of further collisions [2]. Collisions may degrade the performance of a spacecraft or even fragment one if they involve enough energy [3]. Multi-object tracking algorithms play a significant role in space object tracking [4,5].

The catalog size which future space objects tracking systems should be able to handle is more than 100,000 resident space objects [6]. A possible approach to tracking space debris is by the use of a network with regional nodes. Multi-nodal systems which consist of multiple agents with sensing, processing, and communication capabilities are widely used in multiple object tracking systems. Information fusion methods used in multi-node systems fall into three categories: distributed, centralized, and hierarchical. The Joint Space Operations Center (JSpOC) currently employs the centralized information processing approach because it can provide the most accurate estimation [6]. As the space debris tracking system has more nodes, the computation load of centralized systems becomes very challenging. Distributed information fusion can be used as an alternative solution [7]. The main advantages of distributed systems include: the system is reliable (i.e., resilient to failures),

the network topology can be time-varying or unknown, and the processing is scalable with the number of nodes [8].

A *consensus* approach can be used in a distributed system to achieve global information fusion over the network [9,10]. Consensus has emerged as a useful tool for parameter estimation and distributed information fusion [11,12]. The logic behind consensus is to reach a global average by allowing each node to perform iterative local averages [13]. Global average and local average are achieved with information fusion over the whole network and neighboring nodes, respectively. The information can be propagated throughout the entire network with such repeated local information fusion [14,15]. The computation load for each node only depends on the number of its neighboring nodes. The calculation for each node does not increase dramatically with the number of nodes in the whole network [16,17]. A consensus approach for multi-target tracking with labeled random finite set (RFS) filtering was presented in [18]. The *Kullback–Leibler average* for probability densities can be used to achieve consensus for distributed estimation in a general state space model [19].

The scenarios presented in most of the information fusion literature have the same observation volume, with objects staying inside the observation volume all the time. However, space debris does not stay inside the observation volume except for those in geostationary equatorial orbit. Normal multi-object filtering methods do not provide estimation for objects outside the observation volume, and fusion among sensors with different observation volumes fails to provide accurate results.

Multi-object tracking involves the estimation of an unknown number of objects and their trajectories that are time-varying from noisy observations. The RFS approach has been widely used in multi-object tracking. The multi-object state is modeled as an RFS, and a systematic treatment of a multi-object system is provided [20,21]. Due to the complexity of the Bayes filter, the probability hypothesis density (PHD) [22,23], cardinalized PHD (CPHD) [24,25], and multi-Bernoulli filters [26,27] have been proposed as approximations. An analytic solution, the δ -generalized labeled multi-Bernoulli (δ -GLMB) filter and its efficient implementation were proposed in [28,29], respectively. Another efficient implementation with joint prediction and update and Gibbs sampling is detailed in [30]. Marginalized δ -GLMB ($M\delta$ -GLMB) filter and labeled multi-Bernoulli (LMB) filter were proposed in [31,32], respectively, as two efficient approximations.

Most SSA literature treats the core elements of data association, detection, and tracking separately. An integrated approach to all these problems can limit the information loss and improve tracking performance. The δ -GLMB filter can provide object trajectories in the presence of missed detection, clutter, and association uncertainty [28]. It is an exact closed-form solution to multi-object Bayes recursion. The number of components grows without bound in time because of the explicit data associations in the filter. $M\delta$ -GLMB is an efficient approximation to the δ -GLMB filter which preserves both the cardinality distribution and the PHD of the posterior [31]. Moreover, $M\delta$ -GLMB densities are algebraically closed under Kullback–Leibler averaging, which makes $M\delta$ -GLMB filtering a good choice in distributed processing using a consensus approach [18].

The space debris tracking problem has been an active research area [33–40]. A space object tracking approach with CPHD filtering and a measurement-based birth model was presented in [41]. A labeled multi-Bernoulli filter for space object tracking was proposed in [4]. GM-CPHD filtering and a consensus algorithm were used in [42] to achieve global space object tracking. A consensus algorithm was used for distributed information fusion. Even though estimation performance was promising, object trajectories were not available. Preliminary results to track space debris with consensus LMB filtering was proposed in [40], which can only provide state estimation for targets inside of the observation volume and fuse information from sensors with the same observation volume.

This paper proposes a consensus $M\delta$ -GLMB approach for space debris tracking. Sensors can have the same or different observation volumes. Iterative consensus is performed among neighboring nodes to achieve a global average over the sensor network, which makes the fusion scalable with respect to the number of nodes. The topology of the sensor network can be time-varying. The network is resilient to failure since there is no coordination node. In order to achieve space object tracking within a single

framework, the software Turboprop [43] and UKF are used to approximate the transition density of space objects. Turboprop has a package of functions which can be used to calculate the trajectories of space debris.

The main contribution of this paper is the distributed fusion algorithm for sensors with different observation volumes. Most fusion methods assume that all sensors have the same observation volume or that targets stay in the combined observation volume all the time. When targets leave the observation volume, normal algorithms cannot keep the estimation of targets because there are no measurements available. In this paper, the detection probability outside of the observation volume is set to zero. The expectation maximization (EM) algorithm is used to approximate the densities across the observation volumes. Since there are no measurements for targets outside the observation volume, the estimation is essentially the prediction. Then, the fusion among sensors with different observation volumes can be performed in the same way as fusion among sensors with the same observation volume.

The paper is organized as follows. Background on labeled RFS and the network model are described in Section 2. The $M\delta$ -GLMB filter recursion is presented in Section 3. The space debris dynamic model is provided in Section 4. Section 5 details the estimation for objects when they are outside the observation volume. Section 6 details the consensus with $M\delta$ -GLMB filtering. Numerical results are presented in Section 7, and concluding remarks are given in Section 8.

2. Background

2.1. Notation

In this paper, the Kronecker delta that takes arbitrary arguments is denoted by:

$$\delta_Y(X) \triangleq \begin{cases} 1, & \text{if } X = Y, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

The inclusion function is denoted by:

$$1_Y(X) \triangleq \begin{cases} 1, & \text{if } X \subseteq Y, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

The standard inner product notation is denoted by $\langle f, g \rangle \triangleq \int f(x)g(x)dx$, and the exponential notation $h^X \triangleq \prod_{x \in X} h(x)$, where h is a real-valued function. The weighting operator \odot for a given probability density function (PDF) p and a scalar α is defined as:

$$(\alpha \odot p)(x) \triangleq \frac{[p(x)]^\alpha}{\langle p^\alpha, 1 \rangle}. \quad (3)$$

2.2. Network Model

The nodal network can be denoted by a directed graph $\mathcal{G} = (\mathcal{M}, \mathcal{A})$. \mathcal{M} denotes the set of nodes and $\mathcal{A} = \mathcal{M} \times \mathcal{M}$ denotes the connections among nodes. $(i, j) \in \mathcal{A}$ if node j can receive data from node i . $\mathcal{M}^{(j)} \triangleq \{i \in \mathcal{M} : (i, j) \in \mathcal{A}\}$ represents the set of neighbors for each node $j \in \mathcal{M}$ (including j itself).

Every node in the network has the same importance and performs the same actions: gathers measurements, carries out local computation, and exchanges information with neighbors. The fusion is performed among its own information and information from its neighbors.

2.3. Labeled RFS and Bayesian Multi-Object Filtering

An RFS is a finite set valued random variable. A unique label $\ell \in \mathbb{L} = \{\alpha_i : i \in \mathbb{N}\}$ is added to the dynamic state $x \in \mathbb{X}$ to incorporate the identity of objects, where \mathbb{N} is the set of positive integers and the α_i 's are distinct. Labels for objects are ordered pairs of integers $\ell = (k, i)$, where k is the time of object birth and $i \in \mathbb{N}$ is the index to distinguish objects born at the same time. $\mathbb{L}_k = \{k\} \times \mathbb{N}$ is used to denote the label space for objects born at time k . Then, the new object born at time k has the state

$\mathbf{x} \in \mathbb{X} \times \mathbb{L}_k$. The label space can be constructed recursively by $\mathbb{L}_{0:k} = \mathbb{L}_{0:k-1} \cup \mathbb{L}_k$. The abbreviation $\mathbb{L} \triangleq \mathbb{L}_{0:k}$ is used for compactness.

Let $\mathcal{L} : \mathbb{X} \times \mathbb{L} \rightarrow \mathbb{L}$ denote the projection $\mathcal{L}((x, \ell)) = \ell$, then the function $\Delta(\mathbf{X}) \triangleq \delta_{|\mathbf{X}|}(|\mathcal{L}(\mathbf{X})|)$ is called the *distinct label indicator*, which means that \mathbf{X} has the same cardinality as its labels $\mathcal{L}(\mathbf{X}) = \{\mathcal{L}(\mathbf{x}) : \mathbf{x} \in \mathbf{X}\}$. The labeled RFS can be unlabeled by discarding the labels. Let the labeled RFS distribution be denoted by $\pi(\{(x_1, \ell_1), \dots, (x_n, \ell_n)\})$. Then, the unlabeled RFS is distributed according to:

$$\pi(\{x_1, \dots, x_n\}) = \sum_{(\ell_1, \dots, \ell_n) \in \mathbb{L}^n} \pi(\{(x_1, \ell_1), \dots, (x_n, \ell_n)\}).$$

The cardinality distribution (the distribution of the number of objects) of a labeled RFS is the same as its unlabeled version. More information about labeled RFSs can be found in [28,29].

Assume that there are $N(k)$ object states $\mathbf{x}_{k,1}, \dots, \mathbf{x}_{k,N(k)}$ with state space $\mathbb{X} \times \mathbb{L}$ at time k , and $M(k)$ measurements $z_{k,1}, \dots, z_{k,M(k)}$ with observation space \mathbb{Z} . Then, the set of objects and observations are treated as the *multi-object state* and *multi-object observation*, respectively:

$$\begin{aligned} \mathbf{X}_k &= \{\mathbf{x}_{k,1}, \dots, \mathbf{x}_{k,N(k)}\}, \\ Z_k &= \{z_{k,1}, \dots, z_{k,M(k)}\}. \end{aligned}$$

Let $\pi_{k|k-1}$ denote the *multi-object prediction density* and $\pi_k(\cdot|Z_k)$ denote the *multi-object filtering density* at time k . Then, the object density is propagated by the *multi-object Bayes filter* in time according to:

$$\pi_{k|k-1}(\mathbf{X}_k) = \int \mathbf{f}_{k|k-1}(\mathbf{X}_k|\mathbf{X}_{k-1})\pi_{k-1}(\mathbf{X}_{k-1}|Z_{k-1})\delta\mathbf{X}_{k-1}, \quad (4)$$

$$\pi_k(\mathbf{X}_k|Z_k) = \frac{g_k(Z_k|\mathbf{X}_k)\pi_{k|k-1}(\mathbf{X}_k)}{\int g_k(Z_k|\mathbf{X}_k)\pi_{k|k-1}(\mathbf{X}_k)\delta\mathbf{X}'} \quad (5)$$

where $\mathbf{f}_{k|k-1}(\cdot|\cdot)$ is the *multi-object transition density* to time k and $g_k(\cdot|\cdot)$ is the *multi-object likelihood function* at time k . The underlying models of object births, deaths, and motions are encapsulated in the multi-object transition density, while the underlying models for detections and false alarms are encapsulated in the multi-object likelihood function.

For convenience, we omit the references to the time index k and denote $g \triangleq g_k$, $\mathbf{f} \triangleq \mathbf{f}_{k|k-1}$, $\pi_+ \triangleq \pi_{k|k-1}$, $\pi \triangleq \pi_k$, $\mathbb{B} \triangleq \mathbb{L}_k$, $\mathbb{L}_+ \triangleq \mathbb{L} \cup \mathbb{B}$.

2.4. δ -Generalized Labeled Multi-Bernoulli

The δ -GLMB density has the form

$$\pi(\mathbf{X}) = \Delta(\mathbf{X}) \sum_{(I, \xi) \in \mathcal{F}(\mathbb{L}) \times \mathcal{E}} \omega^{(I, \xi)} \delta_I(\mathcal{L}(\mathbf{X})) [p^{(\xi)}]^\mathbf{X}, \quad (6)$$

where each I represents a set of track labels, each ξ represents a history of association maps, and \mathcal{E} is a discrete space. The pair $(I, \xi) \in \mathcal{F}(\mathbb{L}) \times \mathcal{E}$ is called a *hypothesis* and the associated weight $\omega^{(I, \xi)}$ means the probability of the hypothesis. $p^{(\xi)}$ is the density of the kinematic state [28,29].

3. The $M\delta$ -GLMB Filter Recursion

$M\delta$ -GLMB filter is an approximation to the δ -GLMB filter. It can be interpreted as performing a marginalization over the association histories. The $M\delta$ -GLMB filter has the same cardinality distribution and PHD as the δ -GLMB filter [31].

An $M\delta$ -GLMB density corresponding to the δ -GLMB density in (6) is of the form

$$\pi(\mathbf{X}) = \Delta(\mathbf{X}) \sum_{I \in \mathcal{F}(\mathbb{L})} \omega^{(I)} \delta_I(\mathcal{L}(\mathbf{X})) [p^{(I)}]^\mathbf{X}, \quad (7)$$

where

$$\omega^{(I)} = \sum_{\xi \in \Xi} \omega^{(I, \xi)}, \quad (8)$$

$$p^{(I)}(x, \ell) = 1_I(\ell) \frac{1}{\omega^{(I)}} \sum_{\xi \in \Xi} \omega^{(I, \xi)} p^{(\xi)}(x, \ell). \quad (9)$$

A $M\delta$ -GLMB is a special case of the δ -GLMB, and it is completely characterized by the parameter set $\pi = \{(\omega^{(I)}, p^{(I)}) : I \in \mathcal{F}(\mathbb{L})\}$. The $M\delta$ -GLMB prediction and update are outlined below.

3.1. $M\delta$ -GLMB Prediction

For the current time and a given multi-object state \mathbf{X} , each state $(x, \ell) \in \mathbf{X}$ either survives with probability $p_S(x, \ell)$ and evolves to the next step with new state (x_+, ℓ_+) , or dies with probability $1 - p_S(x, \ell)$. The birth density of the new objects at the next time step is

$$\mathbf{f}_B(\mathbf{Y}) = \Delta(\mathbf{Y}) \omega_B(\mathcal{L}(\mathbf{Y})) [p_B]^{\mathbf{Y}}, \quad (10)$$

where p_B and ω_B are given parameters of the birth density.

The multi-object state at the next time step \mathbf{X}_+ includes two parts: the new-born objects and the surviving objects. Supposing that the births are independent of surviving objects and that objects evolve independently of each other, the multi-object transition density is given by [28,29]:

$$\mathbf{f}(\mathbf{X}_+ | \mathbf{X}) = \mathbf{f}_S(\mathbf{X}_+ \cap (\mathbb{X} \times \mathbb{L}) | \mathbf{X}) \mathbf{f}_B(\mathbf{X}_+ - (\mathbb{X} \times \mathbb{L})), \quad (11)$$

where

$$\mathbf{f}_S(\mathbf{W} | \mathbf{X}) = \Delta(\mathbf{W}) \Delta(\mathbf{X}) 1_{\mathcal{L}(\mathbf{X})}(\mathcal{L}(\mathbf{W})) [\Phi(\mathbf{W}; \cdot)]^{\mathbf{X}}, \quad (12)$$

$$\Phi(\mathbf{W}; x, \ell) = \begin{cases} p_S(x, \ell) f(x_+ | x, \ell), & \text{if } (x_+, \ell) \in \mathbf{W}, \\ 1 - p_S(x, \ell), & \text{if } \ell \notin \mathcal{L}(\mathbf{W}). \end{cases} \quad (13)$$

Let $\pi_k = \{(\omega_k^{(I)}, p_k^{(I)}) : I \in \mathcal{F}(\mathbb{L})\}$ denote the $M\delta$ -GLMB multi-object posterior density at time k . The multi-object prediction density is the $M\delta$ -GLMB

$$\pi_{k+1|k}(\mathbf{X}) = \Delta(\mathbf{X}) \sum_{I \in \mathcal{F}(\mathbb{L}_+)} \omega_{k+1|k}^{(I)} \delta_I(\mathcal{L}(\mathbf{X})) [p_{k+1|k}^{(I)}]^{\mathbf{X}}, \quad (14)$$

where

$$\omega_{k+1|k}^{(I)} = \omega_S^{(I)}(I \cap \mathbb{L}_+) \omega_B(I \cap \mathbb{B}_+), \quad (15)$$

$$\omega_S^{(I)}(L) = [\eta_S^{(I)}]^L \sum_{J \supseteq L} [1 - \eta_S^{(I)}]^{J-L} \omega^{(I)}, \quad (16)$$

$$\eta_S^{(I)}(\ell) = \langle p_S(\cdot, \ell), p^{(I)}(\cdot, \ell) \rangle, \quad (17)$$

$$p_{k+1|k}^{(I)}(x, \ell) = 1_{\mathbb{L}_+}(\ell) p_S^{(I)}(x, \ell) + 1_{\mathbb{B}_+}(\ell) p_B(x, \ell), \quad (18)$$

$$p_S^{(I)}(x, \ell) = \frac{\langle p_S(\cdot, \ell) f_{k+1|k}(x | \cdot, \ell), p_{k+1|k}^{(I)}(\cdot, \ell) \rangle}{\eta_S^{(I)}(\ell)}. \quad (19)$$

3.2. $M\delta$ -GLMB Update

Each state $(x, \ell) \in \mathbf{X}$ has the probability $p_D(x, \ell)$ to be detected, generates a measurement with likelihood function $g(z|x, \ell)$, and has a probability $1 - p_D(x, \ell)$ to be mis-detected. The measurement

set Z is a superposition of detected objects and Poisson clutter with intensity function κ .

An association map is denoted by $\theta: \mathbb{L} \rightarrow \{0, 1, \dots, |Z|\}$ such that $\theta(i) = \theta(i') > 0$ implies $i = i'$. The set Θ of all such association maps is used to denote the association space, and the subset of association maps with domain I is denoted by $\Theta(I)$.

Conditional on \mathbf{X} , detections are independent and clutter is independent of detection, then the multi-object likelihood is given by [28,29]:

$$g(Z|\mathbf{X}) = e^{-\langle \kappa, 1 \rangle} \kappa^Z \sum_{\theta \in \Theta(\mathcal{L}(\mathbf{X}))} [\psi_Z(\cdot; \theta)]^{\mathbf{X}}, \tag{20}$$

where

$$\psi_Z(x, \ell; \theta) = \begin{cases} \frac{p_D(x, \ell) g(z_{\theta(\ell)} | x, \ell)}{\kappa(z_{\theta(\ell)})}, & \text{if } \theta(\ell) > 0, \\ 1 - p_D(x, \ell), & \text{if } \theta(\ell) = 0. \end{cases} \tag{21}$$

Given the $M\delta$ -GLMB multi-object prediction density $\pi_{k+1|k} = \{(\omega_{k+1|k}^{(I)}, p_{k+1|k}^{(I)}) : I \in \mathcal{F}(\mathbb{L}_{k+1})\}$, the $M\delta$ -GLMB updated multi-object density at time $k + 1$ is

$$\pi_{k+1}(\mathbf{X}|Z) = \Delta(\mathbf{X}) \sum_{I \in \mathcal{F}(\mathbb{L}_{k+1})} \sum_{\theta \in \Theta(I)} \omega^{(I, \theta)}(Z) \cdot \delta_I(\mathcal{L}(\mathbf{X})) [p^{(I, \theta)}(\cdot | Z)]^{\mathbf{X}}, \tag{22}$$

where

$$\omega_{k+1}^{(I, \theta)}(Z) \propto \omega_{k+1|k}^{(I)} [\eta_Z^{(I, \theta)}]^I, \tag{23}$$

$$\eta_Z^{(I, \theta)}(\ell) = \langle p_{k+1|k}^{(\tilde{\xi})}(\cdot, \ell), \psi(\cdot, \ell; \theta) \rangle, \tag{24}$$

$$p_{k+1}^{(I, \theta)}(x, \ell | Z) = \frac{p_{k+1|k}^{(I)}(x, \ell) \psi_Z(x, \ell; \theta)}{\eta_Z^{(I, \theta)}(\ell)}. \tag{25}$$

The updated $M\delta$ -GLMB density can be denoted by $\pi_{k+1} = \{(\omega_{k+1}^{(I)}, p_{k+1}^{(I)}) : I \in \mathcal{F}(\mathbb{L}_{k+1})\}$. Note that the updated density is not an $M\delta$ -GLMB, but a δ -GLMB. Using (8) and (9), a δ -GLMB can be transformed to an $M\delta$ -GLMB.

The ranked assignment and K-shortest paths algorithms are used in [29] to truncate the multi-object posterior and prediction densities, respectively. The algorithm has a cubic complexity in the number of measurements.

Given the posterior $M\delta$ -GLMB density $\pi = \{(\omega^{(I)}, p^{(I)}) : I \in \mathcal{F}(\mathbb{L})\}$, a tractable suboptimal multi-object estimation can be derived as follows:

- (1) Determine the maximum a posteriori cardinality estimation N^* from

$$Pr(|X| = n) = \sum_{I \in \mathcal{F}(\mathbb{L})} \delta_n(|I|) \omega(I); \tag{26}$$

$$N^* = \arg \max_n Pr(|X| = n). \tag{27}$$

- (2) Among the label set with cardinality N^* , determine the label set I^* with highest weight $\omega(I)$:

$$I^* = \arg \max_I \omega(I). \tag{28}$$

- (3) Finally, determine the expected values of the kinematic states X^* from $p(x, \ell; I^*)$:

$$X^* = \{(x^*, \ell^*) : \ell^* \in I^*, x^* = \arg \max_x p(x, \ell; I^*)\}. \tag{29}$$

4. Space Debris Dynamic Model

The tracking algorithm needs a dynamic model of the debris and an observation model to track objects. The dynamic model is represented by the transition density function $f_{k|k-1}(\mathbf{X}_k|\mathbf{X}_{k-1})$. It is very difficult to calculate the Markov transition density function of space objects because the space dynamic model is much more complicated than a constant velocity or a constant turn model. However, we can approximate the transition density function of space debris with the help of the software Turboprop and the unscented transform [44]. Turboprop has a package of functions to calculate the trajectories of space objects, and can be called as a function in MATLAB. The elements in Turboprop include: an Earth orientation and atmospheric drag model; the lunar gravity models LP100K, GLGM-2, and LP150Q; the Earth gravity models JGM-3, GGM02C and WGS-84; JPL planetary ephemerides DE403 and DE405; and a solar radiation pressure model.

The Jet Propulsion Laboratory Development Ephemeris (JPL DE) are generally created to support spacecraft missions to the planets. JPL DE designates a series of models consisting of representations of accelerations, velocities, and positions of major Solar System bodies. The acceleration caused by the solar radiation pressure is

$$\ddot{\mathbf{r}}_{SRP} = p_{SR} c_R \frac{A}{m} \frac{\bar{\mathbf{r}}}{r}, \quad (30)$$

where p_{SR} is the pressure of solar radiation in Pa. c_R is the solar radiation coefficient and taken as 1.5. m is the mass of the space debris in kilograms. A is the cross-sectional area of the space debris facing the Sun in square meters. m and A are 0.05 kg and 0.01 m², respectively. $\bar{\mathbf{r}}$ represents the vector from the center of the Sun to the space debris. The mass concentration model is a gravity field, with the total acceleration determined by a series of point masses. This paper takes the Sun, the Earth, Venus, Jupiter, and the Moon into consideration.

The unscented transform is used in the nonlinear projection of mean and covariance estimations. The unscented transform approximates the probability density function (PDF) by a bunch of sigma points. Suppose that each single object density $p(x)$ is a Gaussian mixture of the form $\sum_{i=1}^N \omega_i \mathcal{N}(x; m_i, P_i)$ and the Gaussian item is n -dimensional. Then, the state of space debris is propagated as shown in Table 1.

Table 1. State propagation of space debris.

<ul style="list-style-type: none"> • input: $\{\omega_i, m_i, P_i\}_{i=1}^N$ • output: $\{\omega_i^+, m_i^+, P_i^+\}_{i=1}^N$
for $i = 1 : N$ <ol style="list-style-type: none"> (1) $2n + 1$ weighted sigma points $S^{(j)} = \{(W^{(j)}, \chi^{(j)})\}$ are chosen to represent the Gaussian. (2) every sigma point is propagated with the help of Turboprop. (3) the predicted Gaussian is reconstructed with the predicted sigma points. end

The sigma-points are chosen as follows:

$$\begin{aligned} \chi^{(0)} &= m, & W^{(0)} &= \frac{\kappa}{n + \kappa}, & j &= 0, \\ \chi^{(j)} &= m + (\sqrt{(n + \kappa)P})_j, & W^{(j)} &= \frac{1}{2(n + \kappa)}, & j &= 1, \dots, n, \\ \chi^{(j)} &= m - (\sqrt{(n + \kappa)P})_j, & W^{(j)} &= \frac{1}{2(n + \kappa)}, & j &= n + 1, \dots, 2n, \end{aligned}$$

where n represents the dimension of the state vector (6 in this paper), $\sum_{j=0}^{2n} W^{(j)} = 1$, $(\sqrt{(n + \kappa)P})_j$ is the j th row or column of the square root $\sqrt{(n + \kappa)P}$, κ is the scaling parameter (2 in this paper).

5. Estimation of Objects Outside the Observation Volume

Most of the literature in object tracking assumes that objects do not go outside of the observation volume. This is not the case for space debris tracking. Filters with no special consideration of detection probability setting cannot provide estimation for objects outside the observation volume because there are no measurements available. However, since the observation model only affects the update and not the prediction, the estimation of objects can still be provided because the prediction is always available. Aiming to provide estimation for the whole area, we set the detection probability to zero for areas outside of the observation volume.

We assume that the single object density is represented by a Gaussian mixture (GM). Since UKF is used in this paper for the space debris transition function, each Gaussian item is represented by a bunch of sigma-points. If all sigma-points are inside or outside the observation volume, the detection probability for this Gaussian item is set to p_{Din} and p_{Dout} , respectively. p_{Din} is the detection probability inside the observation volume and p_{Dout} is the detection probability outside the observation volume. Otherwise, the EM algorithm is used to build a new GM to approximate the current Gaussian item.

An illustration is shown in Figure 1.

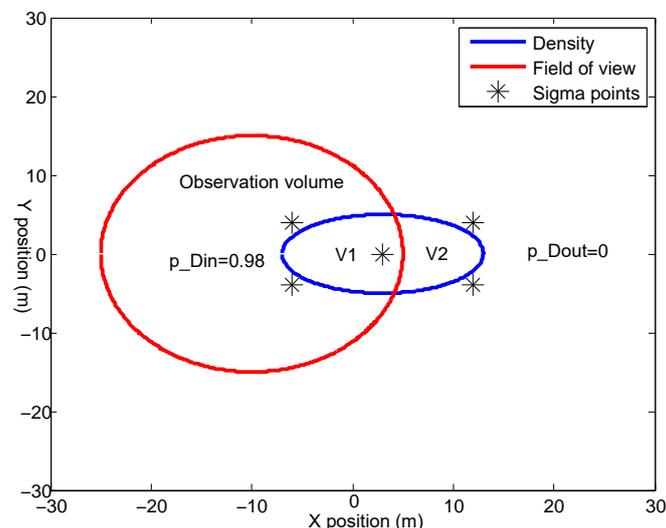


Figure 1. The predicted probability density function is shown by a Gaussian mixture (GM). The blue circle is one Gaussian item. V1 represents the part inside the observation volume, and V2 represents the outside part.

The predicted probability density function is shown by a GM. The blue circle is one Gaussian item. V1 represents the part inside the observation volume, and V2 represents the outside part. * represents the sigma points obtained by the unscented transform. $p_{Din} = 0.98$ is an example of the detection probability inside the observation volume. Since there is no measurement from outside the observation volume, the detection probability is set to $p_{Dout} = 0$. The EM algorithm is used to build a new GM to represent V1 and V2, respectively.

The EM algorithm is an efficient method to find the maximum-likelihood estimate of the unknown parameters of an underlying distribution from a given data set. Usually, the data set has missing values or is incomplete. Each iteration of the EM algorithm has two procedures: The expectation step (E-step) and the maximization step (M-step). The E-step creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters. In the M-step, the likelihood function is maximized under the assumption that the missing data are known.

If the sigma-points of one Gaussian item are on both sides of the observation edge, then $N = 1000$ points are sampled from this Gaussian distribution. Our experience shows that 1000 points are enough to represent the probability distribution in the experiment. Depending on whether the points

are inside the observation volume or not, these points are divided into two groups: $Points_{in}$ and $Points_{out}$. $Points_{in}$ are the sampled points inside the observation volume and $Points_{out}$ are the sampled points outside the observation volume. The EM algorithm is used afterwards for a Gaussian mixture parameter estimation. Each Gaussian item is represented by two new Gaussian mixtures. That is,

$$J^{(\xi)}(\ell) = \sum_{i=1}^{I^{(\xi)}(\ell)} \omega_i^{(\xi)}(\ell) \mathcal{N}(x; m_i^{(\xi)}(\ell), P_i^{(\xi)}(\ell)) = \sum_{i=1}^{I^{(\xi)}(\ell)} \omega_i^{(\xi)}(\ell) \left(\sum_{h=1}^{H^{(\xi)}(\ell)} \omega_h^{(\xi)}(\ell) \mathcal{N}(x; m_h^{(\xi)}(\ell), P_h^{(\xi)}(\ell)) + \sum_{k=1}^{K^{(\xi)}(\ell)} \omega_k^{(\xi)}(\ell) \mathcal{N}(x; m_k^{(\xi)}(\ell), P_k^{(\xi)}(\ell)) \right). \quad (31)$$

The first GM is to represent V1 and the second GM is to represent V2. An illustration of the method is shown in Figure 2.

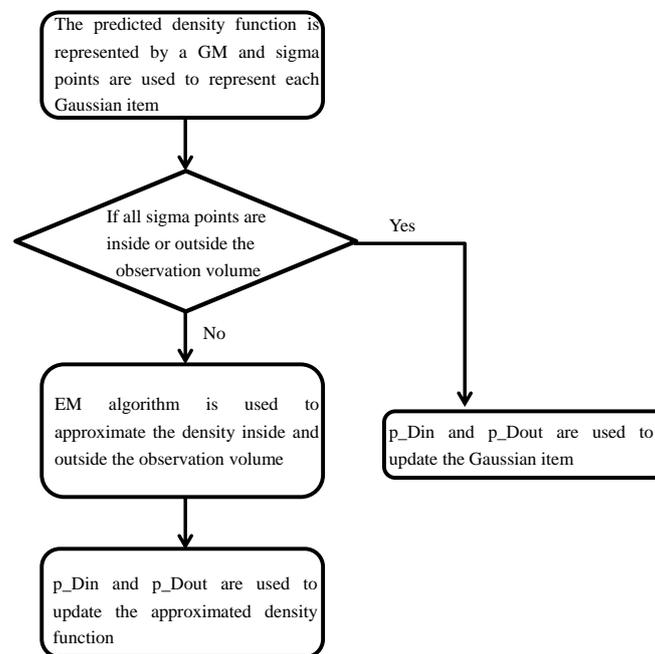


Figure 2. Estimation for objection outside the observation volume. EM: expectation maximization.

The estimation of objects is available when they are outside the observation volume. The proposed consensus $M\delta$ -GLMB can be used for sensors with different observation volumes.

The parameters in the density approximation are the number of Gaussian items, the weights for the items, the covariance matrices, and the means. These parameters are chosen as follows.

(1) The number of Gaussian components:

The number of new Gaussian mixture components is important in the density approximation with the EM algorithm. However, it is very difficult to derive the number of Gaussian components. In practice, we choose a reasonably large number ($J_{max} = 10$ in this paper) as the maximum number of new Gaussian mixture items in the approximation. The EM algorithm is used for situations with a different number of Gaussian components ($J_{min} = 1, \dots, J_{max} = 10$). Then, the Bayesian information criterion (BIC) is used to determine the best number of Gaussian components. The BIC is a criterion

for model selection among a finite set of models [45]. The model with the lowest BIC is preferred. The BIC is defined as follows:

$$BIC(J, \theta, P) = -2 \log L(\theta, J|P) + M \ln(N), \quad (32)$$

where J is the number Gaussian items set for the approximation, θ is the estimated parameter values that maximize the likelihood function, P is the sampled data, $L(\cdot)$ is the maximized value of the likelihood function, M is the number of parameters estimated (6 in this paper), and N is the number of sampled points (1000 in this paper).

(2) The weights for the Gaussian items:

The weights can be the same and set as $W_j = \frac{1}{J}$.

(3) The covariance and the means:

The means of the Gaussian position components can be set by uniformly sampling in the observation volume. The means of the Gaussian velocity components can be set to zero. The initial covariance is chosen by experience, which is the the same covariance matrix of the Gaussian item that was sampled from.

6. Distributed Information Fusion with Consensus for $M\delta$ -GLMB

After each node finishes the state estimation of space debris, a distributed information fusion can be performed to achieve better results. The consensus algorithm is used in this paper to achieve distributed information fusion. This section shows that the $M\delta$ -GLMB is algebraically closed under Kullback–Leibler averaging (KLA) (i.e., the KLA of $M\delta$ -GLMB is also $M\delta$ -GLMB) [18]. The closed form expression for KLA of the $M\delta$ -GLMB is derived and used for consensus fusion of the $M\delta$ -GLMB posterior density [18].

6.1. Consensus for $M\delta$ -GLMB Filtering

The assumption that the estimation from each node in a multi-sensor network is independent or that the correlations among them are known is wrong. The correlation is commonly caused by data incest. Data incest happens when the same information takes several paths from the other sensors to the fusion center and the raw measurements are inadvertently used multiple times. Centralized information fusion is carried out by the Bayes solution:

$$f(X|Z_a \cup Z_b) = \frac{f(Z_a \cup Z_b|X)f(X)}{f(Z_a \cup Z_b)}. \quad (33)$$

The result can be carried out as follows if the information is treated as being independent:

$$f(X|Z_a \cup Z_b) = \frac{f(Z_a|X)f(Z_b|X)f(X)}{f(Z_a)f(Z_b)}. \quad (34)$$

Since there is a common measurement history and common process noise, the information from each node are not independent from each other. Data incest effects can be effectively counteracted by the consensus algorithm.

Given labeled multi-object densities $\pi^{(i)}$ on $\mathcal{F}(\mathbb{X} \times \mathbb{L})$, $i \in \mathcal{I}$, and the normalized non-negative weights $w^{(i)}$, $i \in \mathcal{I}$, the weighted KLA is defined by

$$\bar{\pi} \triangleq \arg \min_{\pi} \sum_{i \in \mathcal{I}} w^{(i)} D_{KL}(\pi || \pi^{(i)}), \quad (35)$$

where

$$D_{KL}(\pi || \pi^{(i)}) \triangleq \int \pi(\mathbf{X}) \log\left(\frac{\pi(\mathbf{X})}{\pi^{(i)}(\mathbf{X})}\right) \delta\mathbf{X} \tag{36}$$

is the Kullback–Leibler Divergence (KLD) of $\pi^{(i)}$ from π [20,22]. The *set integral* is used in the integral for any function f on $\mathcal{F}(\mathbb{X} \times \mathbb{L})$:

$$\int f(\mathbf{X}) \delta\mathbf{X} = \sum_{i=0}^{\infty} \frac{1}{i!} \sum_{(\ell_1, \dots, \ell_i) \in \mathbb{L}^i} \int f(\{(x_1, \ell_1), \dots, (x_i, \ell_i)\}) d(x_1, \dots, x_i). \tag{37}$$

Given normalized non-negative weights $w^{(i)}, i \in \mathcal{I}$ and labeled multi-object densities $\pi^{(i)}, i \in \mathcal{I}$, then the weighted KLA in (35) is

$$\bar{\pi} = \bigoplus_{i \in \mathcal{I}} (w^{(i)} \odot \pi^{(i)}), \tag{38}$$

where

$$\bigoplus_{i \in \mathcal{I}} (w^{(i)} \odot \pi^{(i)}) \triangleq \frac{\prod_{i \in \mathcal{I}} [\pi^{(i)}(\mathbf{X})]^{w^{(i)}}}{\int \prod_{i \in \mathcal{I}} [\pi^{(i)}(\mathbf{X})]^{w^{(i)}} \delta\mathbf{X}} \tag{39}$$

is the normalized weighted geometric mean [18]. It was shown in [13] that independent, identically distributed cluster and Poisson RFSs are algebraically closed under KL averaging. The $M\delta$ -GLMB family is also algebraically closed under KL averaging.

The following result shows that the KLA of the $M\delta$ -GLMB densities is also a $M\delta$ -GLMB density [18].

Given the $M\delta$ -GLMB densities $\pi^{(i)} = \{(\omega_i^{(I)}, p_i^{(I)}) : I \in \mathcal{F}(\mathbb{L})\}, i \in \mathcal{I}$ and normalized non-negative weights $w^{(i)}, i \in \mathcal{I}$, the KLA, and hence the normalized weighted geometric mean, is an $M\delta$ -GLMB given by [18]

$$\bar{\pi} = \bigoplus_{i \in \mathcal{I}} (w^{(i)} \odot \pi^{(i)}) = \{(\bar{\omega}^{(L)}, \bar{p}^{(L)}) : L \in \mathcal{F}(\mathbb{L})\}, \tag{40}$$

where

$$\bar{\omega}^{(L)} = \frac{\prod_{i \in \mathcal{I}} (\omega_i^{(L)})^{w^{(i)}} \left[\int \prod_{i \in \mathcal{I}} (p_i^{(L)}(x, \cdot))^{w^{(i)}} \right]^L}{\sum_{J \subseteq \mathbb{L}, i \in \mathcal{I}} \prod_{i \in \mathcal{I}} (\omega_i^{(J)})^{w^{(i)}} \left[\int \prod_{i \in \mathcal{I}} (p_i^{(J)}(x, \cdot))^{w^{(i)}} \right]^J}, \tag{41}$$

$$\bar{p}^{(L)}(\cdot, \ell) = \frac{\prod_{i \in \mathcal{I}} (p_i^{(L)}(\cdot, \ell))^{w^{(i)}}}{\int \prod_{i \in \mathcal{I}} (p_i^{(L)}(x, \ell))^{w^{(i)}} dx}. \tag{42}$$

The component $(\bar{\omega}^{(L)}, \bar{p}^{(L)}(\cdot))$ of the KLA $M\delta$ -GLMB can be rewritten as

$$\bar{\omega}^{(L)} \propto \prod_{i \in \mathcal{I}} (\omega_i^{(L)})^{w^{(i)}} \left[\int \prod_{i \in \mathcal{I}} (p_i^{(L)}(x, \cdot))^{w^{(i)}} \right]^L, \tag{43}$$

$$\bar{p}^{(L)}(\cdot) = \bigoplus_{i \in \mathcal{I}} (w^{(i)} \odot p_i^{(L)}(\cdot)), \tag{44}$$

where (44) is the Chernoff fusion rule for the single-object PDFs. It can be seen from (43) and (44) that each fused M δ -GLMB component ($\bar{\omega}^{(L)}, \bar{p}^{(L)}(\cdot)$) can be independently calculated, which makes the fusion procedure fully parallelizable.

Given a node network \mathcal{N} with multi-object probability densities $\pi^{(i)}$ from each node i , and normalized non-negative weights $w^{(i,j)}$ for node i to nodes $j \in \mathcal{N}^{(i)}$, with $\sum_{j \in \mathcal{N}^{(i)}} w^{(i,j)} = 1$, the KLA over the whole network can be calculated in a scalable and distributed way using the consensus algorithm. Suppose that each node starts with initial PDF $\pi_0^{(i)} = \pi$. The n th consensus can be calculated by

$$\pi_n^{(i)} = \bigoplus_{j \in \mathcal{N}} (w_n^{(i,j)} \odot \pi^{(j)}), \quad (45)$$

where $w_n^{(i,j)}$ is the (i,j) -th entry of the matrix Ω^n . The (i,j) -th entry of the consensus matrix Ω is given by $w^{(i,j)} 1_{\mathcal{N}^{(i)}}(j)$. The consensus algorithm enjoys some good convergence properties. It was shown that if the consensus matrix is doubly stochastic (all rows and columns sum up to 1) and primitive (there exists an integer m that Ω^m has all positive entries), then

$$\lim_{n \rightarrow \infty} w_n^{(i,j)} = \frac{1}{|\mathcal{N}|}. \quad (46)$$

So, with a doubly stochastic and primitive consensus matrix, the global unweighted KLA of the densities over the whole network can be calculated by the consensus iterative of each node as the number of consensus steps tends to infinity [13].

A necessary condition for the consensus matrix to be primitive is that the network is strongly connected, which means that for any pair of nodes $(i,j) \in \mathcal{N}$ there exists a directed path from node i to node j and vice versa. This condition can also be satisfied if $w^{(i,j)} > 0 \forall i \in \mathcal{N}, j \in \mathcal{N}^{(i)}$. The consensus matrix is primitive and doubly stochastic for a undirected network (i.e., whenever node i sends information to node j , it also receives information from node j) if [46]:

$$w^{(i,j)} = \begin{cases} \frac{1}{1 + \max\{|\mathcal{N}^{(i)}|, |\mathcal{N}^{(j)}|\}}, & i \in \mathcal{N}, j \in \mathcal{N}^{(i)} \setminus \{i\}, \\ 1 - \sum_{j \in \mathcal{N}^{(i)} \setminus \{i\}} w^{(i,j)}, & i \in \mathcal{N}, j = i. \end{cases} \quad (47)$$

The global unweighted KLA of the multi-object posterior densities can be achieved by iterative local KLA averaging as the consensus step tends to infinity. In practice, the iteration is stopped at some finite number.

A *Gaussian mixture* (GM) is a typical choice to represent single-object densities. The fusion rule involves multiplication and exponentiation of GMs which generally do not provide a GM. A suitable approximation of the GM exponentiation has to be devised in order to preserve the GM form. For the sake of simplicity, let us consider only two agents (labeled a and b) with Gaussian mixture location densities

$$p^i(x) = \sum_{j=1}^{N_G^i} \alpha_j^i \mathcal{N}(x; \bar{x}_j^i, P_j^i), i = a, b. \quad (48)$$

The fused location PDF is also a Gaussian mixture, and can be approximated as follows:

$$\bar{p}(x) = \frac{\sum_{i=1}^{N_G^a} \sum_{j=1}^{N_G^b} \alpha_{ij}^{ab} \mathcal{N}(x; \bar{x}_{ij}^{ab}, P_{ij}^{ab})}{\sum_{i=1}^{N_G^a} \sum_{j=1}^{N_G^b} \alpha_{ij}^{ab}}, \quad (49)$$

where

$$P_{ij}^{ab} = [\omega(P_i^a)^{-1} + (1 - \omega)(P_j^b)^{-1}]^{-1}, \tag{50}$$

$$\bar{x}_{ij}^{ab} = P_{ij}^{ab}[\omega(P_i^a)^{-1}\bar{x}_i^a + (1 - \omega)(P_j^b)^{-1}\bar{x}_j^b], \tag{51}$$

$$\alpha_{ij}^{ab} = (\alpha_i^a)^\omega (\alpha_j^b)^{1-\omega} \kappa(\omega, P_i^a) \kappa(1 - \omega, P_j^b) \cdot \mathcal{N}(\bar{x}_i^a - \bar{x}_j^b; 0, \frac{P_i^a}{\omega} + \frac{P_j^b}{1 - \omega}), \tag{52}$$

$$\kappa(\omega, P) = \frac{[\det(2\pi P\omega^{-1})]^{\frac{1}{2}}}{[\det(2\pi P)]^{\frac{\omega}{2}}}. \tag{53}$$

The validity of the equations above depends on the cross-products of the different terms in the Gaussian mixture being negligible, which requires that the Gaussian components are well separated. This condition is well-met since a suitable merging step is used to fuse Gaussian components with Mahalanobis distance [47] below a given threshold. The fusion can be easily extended to more nodes by sequentially applying the pairwise fusion (50) and (51).

The other common way to represent a single-object density is by *particles*. The local filtering steps are more resource-demanding than a GM implementation. Moreover, the in-node computation burden is increased compared to GM representation because the information fusion requires additional techniques (e.g., least square estimation, kernel density estimation, or parametric model approaches).

6.2. Discussions on Consensus

Each node in the network performs local prediction and update. They then exchange information with their neighbors. Information fusion with the consensus algorithm is carried out afterwards. Depending on the requirements of the system, the procedure can be repeated N times, which is called N-steps of consensus. An illustration of the system design is shown in Figure 3.

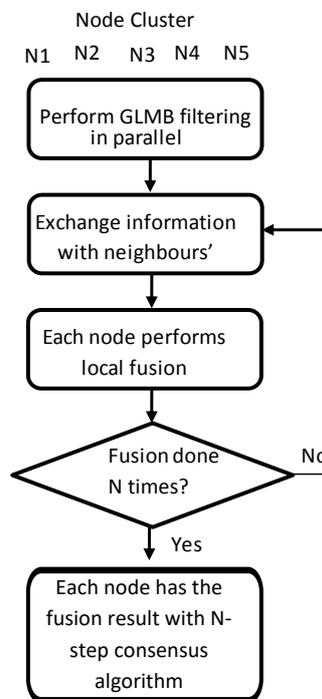


Figure 3. System design.

The network used in this paper is a fully distributed one. Every node performs the same operations: updates its own information, exchanges information with its neighbors, and performs information fusion using the consensus algorithm.

The consensus algorithm achieves global averaging over the whole network with iterative averaging among neighboring nodes for each node. The main disadvantage is the computation complexity. With an increase of network size, the computation load for the whole network and the time it takes to reach global averaging significantly increases. The performance degradation is mainly due to the time it takes for the information from each node to be transferred to the rest of the nodes. In practice, the number of nodes in the whole network is chosen according to the computation power of each node so that the time required for the consensus algorithm is kept under a certain threshold.

It was proved in [48] that the complexity of the centralized $M\delta$ -GLMB filter is linearly related to the number of sensors. Each node in the consensus $M\delta$ -GLMB filtering system has to carry out local prediction and local update. The computation complexity is independent of the number of nodes. The computation complexity of the consensus algorithm for each node only depends on the number of neighboring nodes, and has nothing to do with the total number of nodes in the whole network.

As the number of targets increases, tracking becomes more challenging. Even though the ranked assignment algorithm and K-shortest paths algorithm are used to truncate the posterior densities and prediction densities, respectively, the filtering still has cubic complexity in the number of measurements. An efficient implementation of the GLMB filter by combining the prediction and update into a single step was proposed in [30]. The earlier implementation involves separate truncations in the prediction and update steps. The proposed implementation requires only one truncation procedure for each iteration. Furthermore, an efficient algorithm based on Gibbs sampling for GLMB filtering density truncation was also proposed in [30]. The implementation has a linear complexity in the number of measurements and a quadratic one in the number of objects. The $M\delta$ -GLMB filter used in this paper was proposed in [31]. It is a tractable multi-object density approximation that can capture statistical dependence between objects. It matches the cardinality distribution and the first moment of the labeled multi-object distribution of interest. The performance of $M\delta$ -GLMB filtering is sufficient for the situation in this paper.

The computation load of fusion for each node only depends on the consensus steps and the number of neighbors. For example, if the diameter of a network is 3, then it takes three steps at most for information in any node to be transferred to any other node in this network. As the number of nodes increases, it takes more time to reach the global average. However, the computation for each node at one time is the same, and only depends on the number of neighbors. Large networks will only be limited by the computation capability of each node. Therefore, the system is scalable and the processing load for each node is scalable with respect to the size of the network.

Fully decentralized systems suffer from the problem of synchronization and inconsistency. As the number of nodes increases, it becomes more difficult to achieve synchronization in the system. The effect of communication delays becomes more severe in larger networks. Even if the time delays are small, they can deteriorate the system's performance or even destabilize it. The output synchronization of nonlinear systems with communication time delays was discussed in [49]. A new framework was proposed in [50] to address the consensus with multi-agent systems and the synchronization of complex networks. Coordination can be achieved for the discrete-time delayed systems with linear dynamics [51] and switching topologies [52].

A space debris tracking approach with GM-CPHD and consensus algorithm was presented in [42]. The performance of the approach was shown with an example of 45 clusters of sensors to achieve global tracking of space debris. While the performance was very good, the approach used in this paper has much better results. This is because the $M\delta$ -GLMB filter has a more accurate propagation of the posterior density than the CPHD filter, which makes the $M\delta$ -GLMB filter better able to locate targets. Another disadvantage of the CPHD filter is the "spooky" effect which causes the CPHD filter to temporarily drop tracks which are subjected to missed detections and to declare multiple estimates

for existing tracks in place of the dropped tracks [29]. Further, the CPHD filter cannot provide the trajectories of the targets. The estimates of the target from the CPHD filter are indistinguishable. The $M\delta$ -GLMB filter used in this paper significantly outperforms the CPHD filter used in [42].

This paper presents a scalable solution to the problem of tracking space debris. The space tracking approach employed in this paper is an improvement over existing methods in several aspects. Firstly, the core elements of data association, detection, and tracking are solved with an integrated approach based on labeled RFS in this paper, while most SSA literature treats them separately. Secondly, most SSA methods use a single sensor (e.g., [4,35,41,53]) or a centralized tracking system as in [6] to track space objects. The consensus algorithm used in this paper has better performance than single-sensor methods and has a smaller computation load than a centralized tracking system. Thirdly, the consensus approach used in this paper can provide estimation for targets outside of the observation volume and can be used for information fusion for sensors with different observation volumes, unlike most consensus algorithms (e.g., [13,18,54]) or other distributed information fusion algorithms (e.g., [55,56]) in the current literature which can only be used for sensors with the same observation volume.

The consensus iterate of each node converges to the global unweighted KLA of the multi-object posterior densities as n tends to infinity. In practice, consensus steps N is chosen based on the network size. A methodology to achieve distributed average consensus in finite time was proposed in [57]. The proposed algorithm requires less memory at the cost of a slight increase in the number of steps required for termination. To relax the assumption that the nodes are aware of the upper bound on the network diameter, the authors provide an upper bound on the network diameter which, in the worst case, is twice the actual diameter. Ref. [58] reaches consensus in finite time using only linear iterations. The authors show that finite-time average consensus can always be achieved for undirected networks.

This paper treats the consensus step N as prior information. N is broadcasted to every node before the information fusion. This is certainly possible, but it becomes difficult when the size of the network grows. There are some approaches that do not require continuous data exchange nor knowledge on a global parameter. Ref. [59] reviews some of the main discrete and finite time average consensus implementations, from theoretical and empirical points of view. Three main aspects, namely *computational analysis*, *packet loss resilience analysis*, and *stealth attacks resilience analysis*, are analyzed. The authors of [59] first review synchronous and uniformly sampled approaches which are based on the standard average consensus iteration [60], on the flooding approach [57], or on a repetition of the max-consensus procedure [57]. They then present event-based finite-time average consensus implementations with point-to-point communication capability based on the Converge Cast [61] and on the token-passing approaches [62].

7. Numerical Results

The goal of this paper is to show the efficacy of consensus $M\delta$ -GLMB in tracking space debris. We demonstrate this through three experiments. The first experiment was designed to show that the $M\delta$ -GLMB filter proposed in this paper with special consideration of detection probability setting can provide estimation for objects outside the observation volume. The second experiment was designed to show that consensus fusion can be performed among sensors with different observation volumes. The third experiment was designed to show that the consensus algorithm can significantly improve the tracking performance when all objects are inside the observation volumes of all sensors.

The object state is a vector of position and velocity, $x_k = [p_x \ p_y \ p_z \ v_x \ v_y \ v_z]^T$. The state dynamic model is:

$$x_{k+1} = f(x_k) + w_k.$$

The first step is to build the space debris trajectories with the help of Turboprop. $f(\cdot)$ is the transition density function and $w_k \sim N(\cdot; 0, Q)$ is the process noise with $Q = \text{diag}([\sigma_{px}^2, \sigma_{py}^2, \sigma_{pz}^2, \sigma_{vx}^2, \sigma_{vy}^2, \sigma_{vz}^2])$, $\sigma_{px} = \sigma_{py} = \sigma_{pz} = 1$ km, $\sigma_{vx} = \sigma_{vy} = \sigma_{vz} = 0.01$ km/s. The birth model to

generate the simulation scenario is a labeled multi-Bernoulli RFS with parameters $\pi_B = \{r_B^{(i)}, p_B^{(i)}\}_{i=1}^5$, where $r_B^{(i)} = 0.02$ and $p_B^{(i)}(x) = \mathcal{N}(x; m_B^{(i)}, P_B)$, with

$$\begin{aligned} m_\gamma^{(1)} &= [42, 097.71 \text{ km}, -2221.31 \text{ km}, -3.68 \text{ km}, 0.1620 \text{ km/s}, 3.0703 \text{ km/s}, 6.7817 \text{ km/s}]^T, \\ m_\gamma^{(2)} &= [42, 097.31 \text{ km}, -2097.2 \text{ km}, -756.51 \text{ km}, 0.1625 \text{ km/s}, 2.8851 \text{ km/s}, 1.0501 \text{ km/s}]^T, \\ m_\gamma^{(3)} &= [42, 096.91 \text{ km}, -1719.39 \text{ km}, -1425.14 \text{ km}, 0.1628 \text{ km/s}, 2.3520 \text{ km/s}, 1.9735 \text{ km/s}]^T, \\ m_\gamma^{(4)} &= [42, 098.08 \text{ km}, -2083.03 \text{ km}, 745.34 \text{ km}, 0.1613 \text{ km/s}, 2.8852 \text{ km/s}, -1.0500 \text{ km/s}]^T, \\ m_\gamma^{(5)} &= [42, 098.43 \text{ km}, -1700.09 \text{ km}, 1407.22 \text{ km}, 0.1610 \text{ km/s}, 2.3522 \text{ km/s}, -1.9734 \text{ km/s}]^T, \\ P_\gamma &= \text{diag}([10 \text{ km}, 10 \text{ km}, 10 \text{ km}, 0.01 \text{ km/s}, 0.01 \text{ km/s}, 0.01 \text{ km/s}]). \end{aligned}$$

$r_B^{(i)}$ is the birth intensity, which is the expected number of new objects born at each time step, at each object birth place i . All targets are initially born inside the observation volume of a sensor (not necessarily the same one). The situation where a target is born outside of the observation volume of all sensors and goes from outside to inside of the observation volume can be handled by a measurement-based birth model, which is beyond the scope of this paper. The space object trajectories generated by the birth model in this experiment are very common in real space object tracking scenarios.

Clutter is modeled as a Poisson RFS with intensity $\kappa_k(z) = \lambda_c V u(z)$. λ_c is the average of clutter returns per unit volume, which was 100 in the simulation. $u(\cdot)$ is the uniform density over the observation region, V is the “volume” of the observation region.

All nodes used in this paper were located on the surface of the Earth and rotated with the Earth. The latitudes and longitudes of nodes were

$$L1 = (0^\circ, 0^\circ), L2 = (0^\circ, 30^\circ N), L3 = (0^\circ, 30^\circ S), L4 = (30^\circ E, 0^\circ), L5 = (30^\circ W, 0^\circ),$$

respectively. Node1 was initially on the x - z plane.

The sensor’s observation is a noisy bearing and range vector given by

$$z_k = \begin{bmatrix} \sqrt{p_{x,k}^2 + p_{y,k}^2 + p_{z,k}^2} \\ \arctan\left(\frac{p_{y,k}}{p_{x,k}}\right) \\ \arctan\left(\frac{p_{z,k}}{\sqrt{p_{x,k}^2 + p_{y,k}^2}}\right) \end{bmatrix} + \varepsilon_k, \quad (54)$$

where $\varepsilon_k \sim N(\cdot; 0, R_k)$ with

$$\begin{aligned} R_{k1} &= \text{diag}([\sigma_{r1}^2, \sigma_{\alpha1}^2, \sigma_{\beta1}^2]), \sigma_{r1} = 0.05 \text{ km}, \sigma_{\alpha1} = 0.02^\circ, \sigma_{\beta1} = 0.02^\circ, \\ R_{k2} &= \text{diag}([\sigma_{r2}^2, \sigma_{\alpha2}^2, \sigma_{\beta2}^2]), \sigma_{r2} = 0.22 \text{ km}, \sigma_{\alpha2} = 0.01^\circ, \sigma_{\beta2} = 0.01^\circ, \\ R_{k3} &= \text{diag}([\sigma_{r3}^2, \sigma_{\alpha3}^2, \sigma_{\beta3}^2]), \sigma_{r3} = 0.12 \text{ km}, \sigma_{\alpha3} = 0.02^\circ, \sigma_{\beta3} = 0.01^\circ, \\ R_{k4} &= \text{diag}([\sigma_{r4}^2, \sigma_{\alpha4}^2, \sigma_{\beta4}^2]), \sigma_{r4} = 0.1 \text{ km}, \sigma_{\alpha4} = 0.01^\circ, \sigma_{\beta4} = 0.02^\circ, \\ R_{k5} &= \text{diag}([\sigma_{r5}^2, \sigma_{\alpha5}^2, \sigma_{\beta5}^2]), \sigma_{r5} = 0.15 \text{ km}, \sigma_{\alpha5} = 0.015^\circ, \sigma_{\beta5} = 0.015^\circ. \end{aligned}$$

$\Delta = 40$ s was the sampling period. The detection probability inside the observation volume $p_{Din} = 0.98$. The δ -GLMB filter was capped to 10,000 components.

In order to show statistical significance, results were shown over 100 Monte Carlo trials for the same object trajectories but different, independently generated, clutter and measurement noise realizations. The first experiment showed the efficacy of the algorithm with one realization. The second and third experiments showed with the mean value over 1000 Monte Carlo trials. Typically, the optimal sub-pattern

assignment (OSPA) performance in the RFS area is shown with mean values. The OSPA distance is the sum of two parts: location error and cardinality error. The cardinality error is shown with not just the mean value, but also the standard deviation.

Performance evaluation of the multi-target algorithm is of great practical importance in the design and comparison of tracking systems. The difference between the estimation and true trajectories is very small compared with the motion range of space debris. Therefore, the OSPA metric was used to evaluate the performance with Euclidean distance $p = 2$, and cutoff parameter $c = 20$ km for the second experiment and $c = 1$ km for the third experiment. The OSPA is a mathematically and intuitively consistent metric for multi-object filtering performance evaluation [63]. OSPA distance has two components: localization and cardinality. The total OSPA distance is the sum of the two components. The cut-off parameter c determines the penalty weighting for cardinality errors as opposed to localization errors. The second experiment was mainly designed to show the estimation performance for targets when they go outside the observation volume. As there were no measurements available, the estimation error became larger and larger, and so did the OSPA value. Therefore, $c = 20$ km was used in this experiment. The third experiment was mainly intended to show the efficacy of the consensus algorithm when targets were all inside the combined observation volumes. Since measurements were available all the time, the estimation error was smaller than in the second experiment. So, the cut-off value $c = 2$ km was used in the third experiment.

The consensus algorithm stops at N steps. The information of N should be broadcasted to every node beforehand and treated as prior information. In practice, N is chosen based on the size of the network. This paper shows the estimation performance with 1-step consensus, 2-step consensus, and 3-step consensus.

7.1. Estimation of Objects Outside the Observation Volume

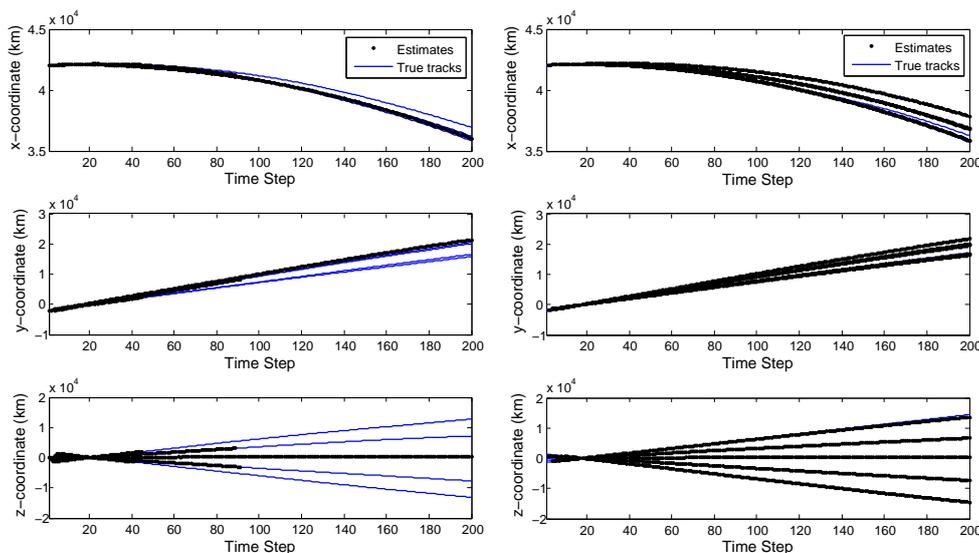
This section demonstrates that estimation can be provided for objects outside the observation volume by setting the detection probability outside the observation volume to zero. Only node1 is used in this section. A fixed number of objects are presented here for simplicity. All objects are inside the observation volume initially, but as time goes on, some objects go outside. The field of view for node1 is

$$V1 = [30,000 \text{ km } 40,000 \text{ km}] \times [-15^\circ 0^\circ] \times [-5^\circ 5^\circ].$$

This yields a sensor volume of 150 deg^2 . Some objects go outside of the observation volume with this setting.

The estimation from the $M\delta$ -GLMB filter with uniform detection probability [13,18,48,54] is shown in Figure 4a. The filtering with uniform detection probability cannot provide estimation for targets when they are outside of the observation volumes. The estimation from $M\delta$ -GLMB with detection probability outside of the observation volume set to zero is shown in Figure 4b.

Since this section is intended to show the estimation of target positions, estimation results from one simulation are shown here, instead of the mean values from all simulations. Note that all the simulations had similar results. It can be seen from Figure 4b that even when the measurements for objects outside the observation volume were not available, estimation could still be provided. In this case, the estimation for objects outside was essentially the prediction. The figure also demonstrates that the estimation became worse for objects outside the observation volume since there were no measurements to update.



(a) $M\delta$ -GLMB estimation with uniform detection probability. Two objects go outside the observation volume at around the 44th time step and another two targets at the 88th time step. Estimation is only available for objects inside the observation volume.

(b) Estimation from marginalized δ -GLMB ($M\delta$ -GLMB) with the detection probability outside observation volume set to zero. Estimation is available for objects outside the observation volume.

Figure 4. Estimation with different detection probability settings.

7.2. Consensus among Nodes with Different Observation Volumes

This subsection demonstrates the effectiveness of information fusion with a consensus algorithm for nodes with different observation volumes. It is complicated to fuse information from nodes with no estimation for objects outside the observation volumes. The fusion algorithm must consider the observation volume overlap, which makes the fusion more difficult when more nodes are involved. With the estimation available for objects outside the observation volume, the fusion algorithm can be performed in the same way as for nodes with the same observation volume.

Five nodes in the same position as node1 with complementary observation volumes are used in this subsection. The observation volumes for nodes were, respectively,

$$\begin{aligned}
 V1 &= [30,000 \text{ km } 40,000 \text{ km}] \times [-15^\circ \ 0^\circ] \times [-5^\circ \ 5^\circ], \\
 V2 &= [30,000 \text{ km } 40,000 \text{ km}] \times [-15^\circ \ 0^\circ] \times [5^\circ \ 15^\circ], \\
 V3 &= [30,000 \text{ km } 40,000 \text{ km}] \times [-15^\circ \ 0^\circ] \times [15^\circ \ 25^\circ], \\
 V4 &= [30,000 \text{ km } 40,000 \text{ km}] \times [-15^\circ \ 0^\circ] \times [-15^\circ \ -5^\circ], \\
 V5 &= [30,000 \text{ km } 40,000 \text{ km}] \times [-15^\circ \ 0^\circ] \times [-25^\circ \ -15^\circ].
 \end{aligned}$$

This ensures that all objects were inside the combined observation volume of all sensors. The OSPA performance of a single-node and 5-node consensus is shown in Figure 5. The cardinality performance of a single-node and 5-node consensus is shown in Figure 6.

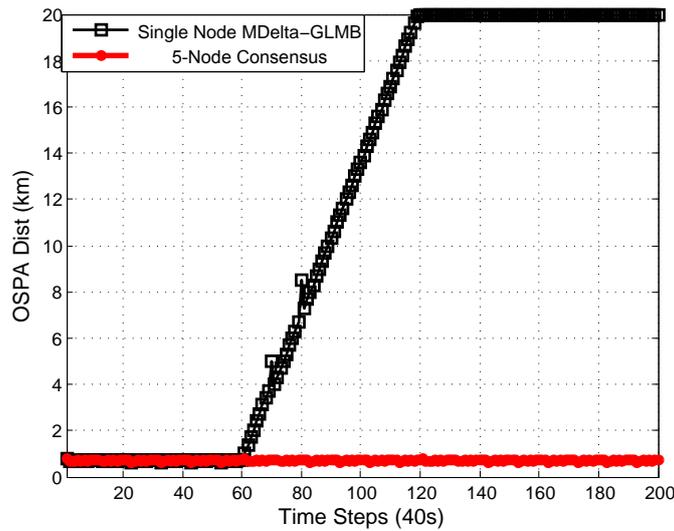


Figure 5. Estimation from single-node and 5-node consensus. Objects are starting to go outside the observation volume from the 60th time step and the performance for a single node becomes worse. Five-node consensus with complementary observation volumes provides accurate estimation for objects as long as they are in at least one node’s observation volume.

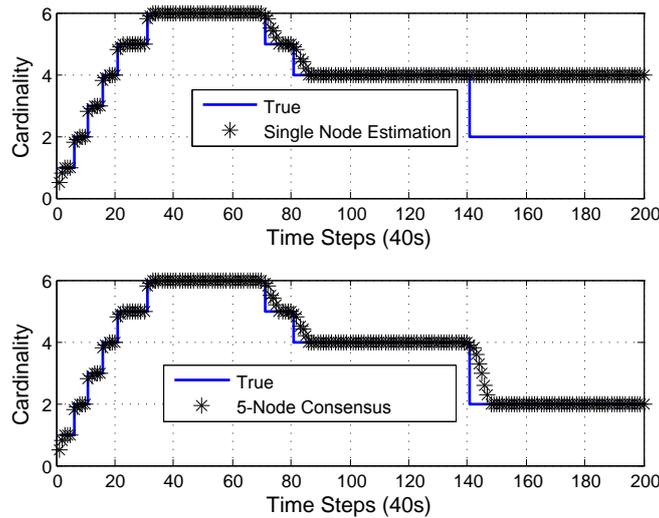


Figure 6. Cardinality performance from single-node and 5-node consensus. Two targets die at the 140th time step. A single node cannot provide cardinality estimation for objects outside the observation volume, while multiple nodes can be used to counteract the lack of observation volume of a single node.

It can be seen that the single-node $M\delta$ -GLMB filter [48] could provide good estimation for objects inside the observation volume. Since there was no update for objects outside the observation volume and only prediction was available, the estimation became worse with time. Two objects disappeared at the 140th time step, at which time they were outside the observation volume. A single node could not provide accurate cardinality estimation for objects outside the observation volume. Since all objects were in the combined observation volume of five nodes all the time, information fusion from five nodes could provide better estimation than a single node for objects outside the observation volume.

Since there was no measurement outside the observation volume, the best we could do is use the prediction model as the final estimation. The estimation performance was subject to the dynamic model which was characterized by the transition density noise. As time progressed, the estimation error became larger and larger without bound until new measurements were captured. The estimation confidence can be illustrated by the covariance matrix.

7.3. Consensus for Nodes with Similar Observation Volumes

This subsection is to show the performance of the consensus algorithm for nodes with similar observation volumes. The angle range of all nodes was 50° , which was large enough for all objects to stay in the observation volume of all nodes all the time. The connection among the nodes is shown in Figure 7. A network with only five nodes was used in this paper to show the efficacy of the proposed method in distributed space debris tracking. A larger-scale network would involve more computation and processing time.

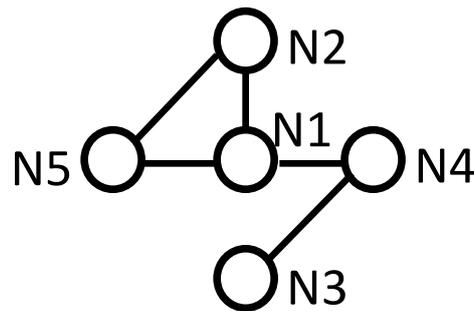


Figure 7. Network topology. Note that this topology is used as an example to show the efficacy of the consensus algorithm. The topology can be time-varying and unknown.

We only show the estimation from node3.

The numerical experimental results from a single sensor [48], 1-step consensus, 5-node consensus, and centralized $M\delta$ -GLMB filtering [48] are shown in Figure 8. Five-node consensus was the fusion procedure for all the information from five nodes. It is shown that the performance of 1-step consensus was better than single-node estimation. Since 5-node consensus uses information from all the nodes, it had better performance than 1-step consensus. Centralized $M\delta$ -GLMB filtering makes use of raw measurements from all sensors, and had the best performance.

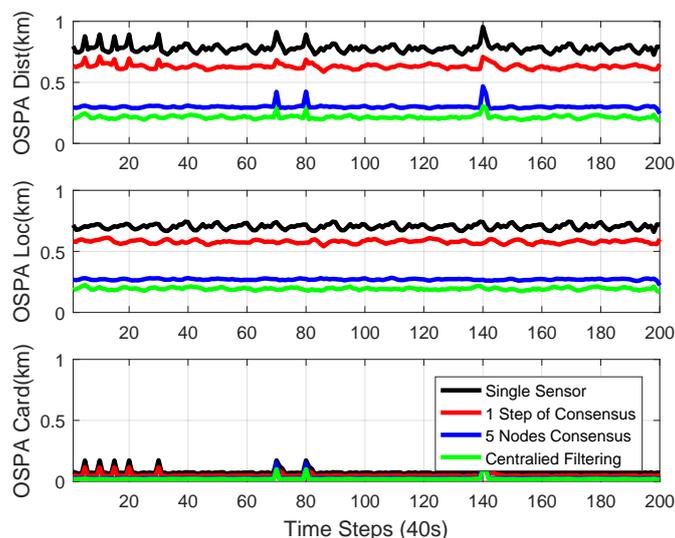


Figure 8. Optimal sub-pattern assignment (OSPA) performance from single sensor, 1-step consensus, 5-node consensus, and centralized $M\delta$ -GLMB filtering.

OSPA performance with different consensus steps is shown in Figure 9. It is shown in the figure that the OSPA performance was better with more consensus steps. With the designed network topology, the 3-step consensus had similar performance to the 5-node consensus. This is because it took 3 steps at most to transfer information from all of the other nodes to node3. After 3 steps of consensus, every

node in the network had similar information, which is the average of the information from all nodes. There was no central node and the processing for each node was only carried among its neighbors, which makes the fusion scalable with respect to the size of the network.

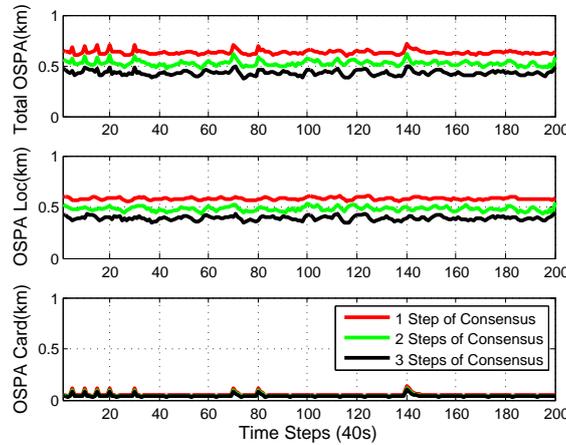


Figure 9. OSPA performance with different consensus steps. More consensus steps yielded better performance.

The mean and standard deviation of the estimated cardinality of single-node $M\delta$ -GLMB, 1-step consensus, 2-step consensus, 3-step consensus, 5-node consensus, and centralized $M\delta$ -GLMB filtering versus time are shown in Figures 10 and 11, respectively. All filters estimated the cardinality accurately. The centralized $M\delta$ -GLMB filtering had the best cardinality estimation performance. Consensus with more steps showed better cardinality estimation.

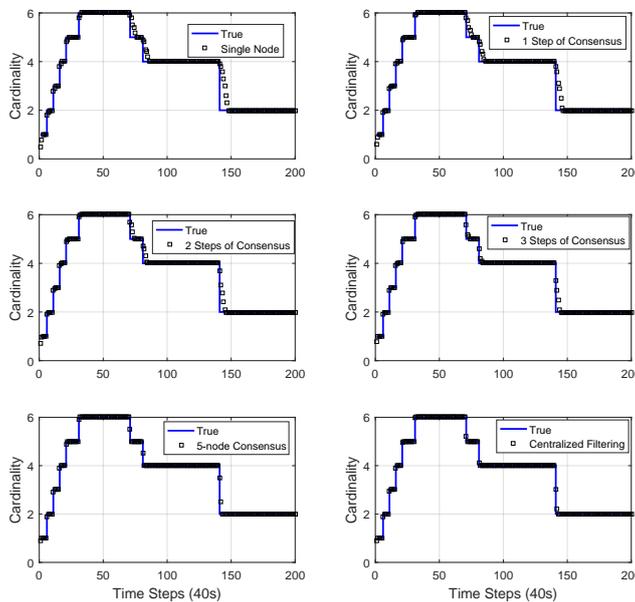


Figure 10. The mean of the estimated cardinality of single-node $M\delta$ -GLMB, 1-step consensus, 2-step consensus, 3-step consensus, 5-node consensus, and centralized $M\delta$ -GLMB filtering versus time.

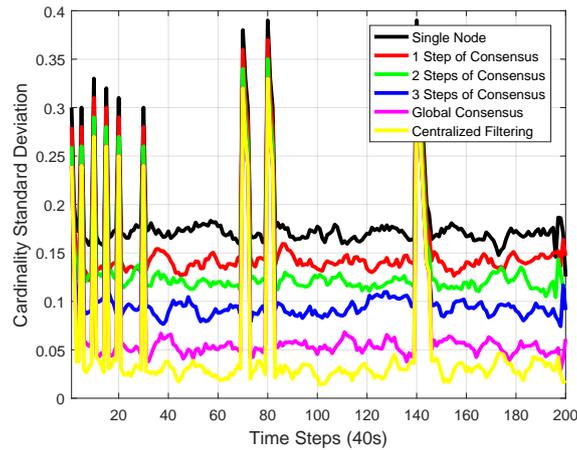


Figure 11. The standard deviation of the estimated cardinality of single-node $M\delta$ -GLMB, 1-step consensus, 2-step consensus, 3-step consensus, 5-node consensus, and centralized $M\delta$ -GLMB filtering versus time.

We can see that there are peaks in the plots in Figures 8, 9 and 11. The peaks happen at the time when there was a change in the number of objects. The change in the number of objects also contributed to the peaks in the cardinality standard deviation estimation. Cardinality estimation had a larger error than when the number of objects was not changing. The peaks in the total OSPA distance are because of the peaks in the cardinality OSPA distance.

The network can also be time-varying because the processing is performed only among neighbors. The time-varying topology is shown in Figure 12, and the performance of the consensus algorithm is shown in Figure 13. Node3 and node4 were in the network from the start to the 60th time step. Node3, node4, and node1 were in the network from the 61st time step to the 130th time step. All five nodes were in the network the rest of the time. Carrying out consensus only among a node’s neighbors makes it possible for the method to handle a network topology that is time-varying.

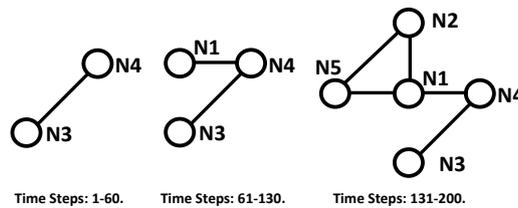


Figure 12. The time-varying topology for different time steps.

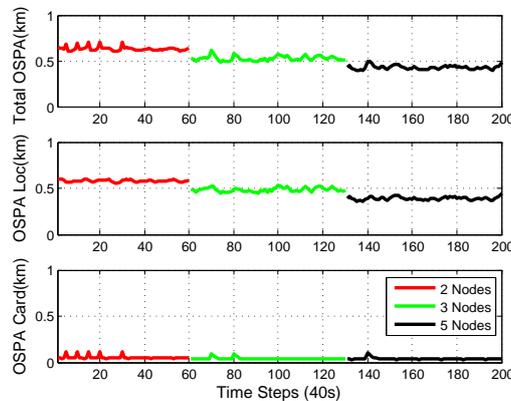


Figure 13. OSPA performance for time-varying topology.

8. Concluding Remarks

This paper presented a consensus algorithm for space debris tracking with labeled RFS. The key innovation lies in the fusion among sensors with different observation volumes, and most importantly, the processing load for each node is scalable with respect to the size of the network. The system is robust because of the avoidance of data incest. A salient feature is that the network topology can be time-varying. The scalability makes the approach appealing for future space object tracking systems as more nodes are connected to the network. Experiments confirm that the proposed algorithm has the potential to be used in distributed space debris tracking systems.

In recent years, there is more need for finite-time average consensus algorithms. Future work will focus on distributed space debris tracking systems with finite-time average consensus methods. Strengths and shortcomings of different consensus algorithms will be presented for different tracking scenarios in future work.

Author Contributions: Conceptualization, B.W. and B.N.; Formal analysis, B.W.; Project administration, B.N.; Resources, B.N.; Software, B.W.; Supervision, B.N.; Writing—original draft, B.W.; Writing—review & editing, B.N.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bobrinsky, N.; Del Monte, L. The space situational awareness program of the European Space Agency. *Cosm. Res.* **2010**, *48*, 392–398. [[CrossRef](#)]
2. Talent, D.; Vilas, F. A space-based concept for a collision warning sensor. In Proceedings of the Space Programs and Technologies Conference, Huntsville, AL, USA, 25–27 September 1990; p. 3871.
3. Jones, B.A.; Doostan, A. Satellite collision probability estimation using polynomial chaos expansions. *Adv. Space Res.* **2013**, *52*, 1860–1875. [[CrossRef](#)]
4. Jones, B.A.; Vo, B.N. A labeled multi-Bernoulli filter for space object tracking. In Proceedings of the 2014 AAS/AIAA Spaceflight Mechanics Meeting, Santa Fe, NM, USA, 26–30 January 2014; pp. 11–15.
5. Jones, B.A.; Doostan, A.; Born, G.H. Nonlinear propagation of orbit uncertainty using non-intrusive polynomial chaos. *J. Guid. Control Dyn.* **2013**, *36*, 430–444. [[CrossRef](#)]
6. National Research Council. *Continuing Kepler's Quest: Assessing Air Force Space Command's Astrodynamics Standards*; National Academies Press: Washington, DC, USA, 2012.
7. Brooks, R.R.; Ramanathan, P.; Sayeed, A.M. Distributed target classification and tracking in sensor networks. *Proc. IEEE* **2003**, *91*, 1163–1171. [[CrossRef](#)]
8. Zou, Y.; Chakrabarty, K. Distributed mobility management for target tracking in mobile sensor networks. *IEEE Trans. Mob. Comput.* **2007**, *6*, 872–887. [[CrossRef](#)]
9. Olfati-Saber, R.; Murray, R.M. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Control* **2004**, *49*, 1520–1533. [[CrossRef](#)]
10. Zhu, S.; Chen, C.; Li, W.; Yang, B.; Guan, X. Distributed optimal consensus filter for target tracking in heterogeneous sensor networks. *IEEE Trans. Cybern.* **2013**, *43*, 1963–1976. [[CrossRef](#)] [[PubMed](#)]
11. Olfati-Saber, R.; Fax, J.A.; Murray, R.M. Consensus and cooperation in networked multi-agent systems. *Proc. IEEE* **2007**, *95*, 215–233. [[CrossRef](#)]
12. Kamal, A.T.; Farrell, J.A.; Roy-Chowdhury, A.K. Information consensus for distributed multi-target tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Singapore, 23–28 June 2013; pp. 2403–2410.
13. Battistelli, G.; Chisci, L.; Fantacci, C.; Farina, A.; Graziano, A. Consensus CPHD filter for distributed multitarget tracking. *IEEE J. Sel. Top. Signal Process.* **2013**, *7*, 508–520. [[CrossRef](#)]
14. Olfati-Saber, R. Kalman-consensus filter: Optimality, stability, and performance. In Proceedings of the 48th IEEE Conference on Decision and Control (CDC) Held Jointly with 2009 28th Chinese Control Conference, Shanghai, China, 15–18 December 2009; pp. 7036–7042.
15. Saber, R.O.; Murray, R.M. Consensus protocols for networks of dynamic agents. In Proceedings of the 2003 American Control Conference, Piscataway, NJ, USA, 4–6 June 2003.

16. Olfati-Saber, R. Distributed Kalman filtering for sensor networks. In Proceedings of the 2007 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, 12–14 December 2007; pp. 5492–5498.
17. Olfati-Saber, R.; Shamma, J.S. Consensus filters for sensor networks and distributed sensor fusion. In Proceedings of the 44th IEEE Conference on Decision and Control, Seville, Spain, 15 December 2005; pp. 6698–6703.
18. Fantacci, C.; Vo, B.-N.; Vo, B.-T.; Battistelli, G.; Chisci, L. Robust fusion for multisensor multiobject tracking. *IEEE Signal Process. Lett.* **2018**, *25*, 640–644. [[CrossRef](#)]
19. Battistelli, G.; Chisci, L. Kullback–Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability. *Automatica* **2014**, *50*, 707–718. [[CrossRef](#)]
20. Mahler, R.P. *Statistical Multisource-Multitarget Information Fusion*; Artech House, Inc.: Norwood, MA, USA, 2007.
21. Mahler, R.P. *Advances in Statistical Multisource-Multitarget Information Fusion*; Artech House: Norwood, MA, USA, 2014.
22. Mahler, R.P. Multitarget Bayes Filtering via First-Order Multitarget Moments. *IEEE Trans. Aerosp. Electron. Syst.* **2003**, *39*, 1152–1178. [[CrossRef](#)]
23. Vo, B.N.; Ma, W.K. The Gaussian mixture probability hypothesis density filter. *Trans. Signal Process.* **2006**, *54*, 4091–4104. [[CrossRef](#)]
24. Mahler, R. PHD filters of higher order in target number. *IEEE Trans. Aerosp. Electron. Syst.* **2007**, *43*, 1523–1543. [[CrossRef](#)]
25. Vo, B.T.; Vo, B.N.; Cantoni, A. Analytic implementations of the cardinalized probability hypothesis density filter. *IEEE Trans. Signal Process.* **2007**, *55*, 3553–3567. [[CrossRef](#)]
26. Vo, B.T.; Vo, B.N.; Cantoni, A. The cardinality balanced multi-target multi-Bernoulli filter and its implementations. *IEEE Trans. Signal Process.* **2009**, *57*, 409–423.
27. Vo, B.N.; Vo, B.T.; Pham, N.T.; Suter, D. Joint detection and estimation of multiple objects from image observations. *IEEE Trans. Signal Process.* **2010**, *58*, 5129–5141. [[CrossRef](#)]
28. Vo, B.T.; Vo, B.N. Labeled random finite sets and multi-object conjugate priors. *IEEE Trans. Signal Process.* **2013**, *61*, 3460–3475. [[CrossRef](#)]
29. Vo, B.N.; Vo, B.T.; Phung, D. Labeled random finite sets and the Bayes multi-target tracking filter. *IEEE Trans. Signal Process.* **2014**, *62*, 6554–6567. [[CrossRef](#)]
30. Vo, B.N.; Vo, B.T.; Hoang, H. An Efficient Implementation of the Generalized Labeled Multi-Bernoulli Filter. *IEEE Trans. Signal Process.* **2016**, *65*, 1975–1987. [[CrossRef](#)]
31. Papi, F.; Vo, B.N.; Vo, B.T.; Fantacci, C.; Beard, M. Generalized Labeled Multi-Bernoulli Approximation of Multi-Object Densities. *IEEE Trans. Signal Process.* **2015**, *63*, 5487–5497. [[CrossRef](#)]
32. Reuter, S.; Vo, B.T.; Vo, B.N.; Dietmayer, K. The labeled multi-Bernoulli filter. *IEEE Trans. Signal Process.* **2014**, *62*, 3246–3260.
33. Faber, W.; Chakravorty, S.; Hussein, I.I. A randomized sampling based approach to multi-object tracking. In Proceedings of the 2015 18th International Conference on Information Fusion (Fusion), Washington, DC, USA, 6–9 July 2015; pp. 1307–1314.
34. Hussein, I.; DeMars, K.; Früh, C.; Jah, M.; Erwin, R. An AEGIS-FISST algorithm for multiple object tracking in space situational awareness. In Proceedings of the AIAA/AAS Astrodynamics Specialist Conference, Minneapolis, MN, USA, 13–16 August 2012.
35. DeMars, K.J.; Hussein, I.I.; Frueh, C.; Jah, M.K.; Scott Erwin, R. Multiple-Object Space Surveillance Tracking Using Finite-Set Statistics. *J. Guid. Control Dyn.* **2015**, *38*, 1741–1756. [[CrossRef](#)]
36. Faber, W.; Chakravorty, S.; Hussein, I.I. Multi-Object Tracking with Multiple Birth, Death, and Spawn Scenarios Using A Randomized Hypothesis Generation Technique (R-FISST). *arXiv* **2016**, arXiv:1603.07684.
37. Hussein, I.; Frueh, C.; Erwin, R.; Jah, M. An AEGIS-FISST algorithm for joint detection, classification, and tracking. In Proceedings of the 2013 AAS/AIAA Astrodynamics Specialist Conference, Hilton Head Island, SC, USA, 11–15 August 2013.
38. Hussein, I.I.; DeMars, K.J.; Früh, C.; Erwin, R.S.; Jah, M.K. An AEGIS-FISST integrated detection and tracking approach to Space Situational Awareness. In Proceedings of the 2012 15th International Conference on Information Fusion, Singapore, 9–12 July 2012.

39. DeMars, K.J.; Hussein, I.I.; Jah, M.K.; Erwin, R.S. The Cauchy-Schwarz divergence for assessing situational information gain. In Proceedings of the 2012 15th International Conference on Information Fusion, Information Fusion (FUSION), Singapore, 9–12 July 2012.
40. Wei, B.; Nener, B. Consensus labeled multi-Bernoulli filtering for distributed space debris tracking. In Proceedings of the 2017 International Conference on Control, Automation and Information Sciences (ICCAIS), Chiang Mai, Thailand, 31 October–1 November 2017.
41. Jones, B.A.; Gehly, S.; Axelrad, P. Measurement-based birth model for a space object cardinalized probability hypothesis density filter. In Proceedings of the AIAA/AAS Astrodynamics Specialist Conference, AIAA SPACE Forum, (AIAA 2014-4311), San Diego, CA, USA, 4–7 August 2014.
42. Wei, B.; Nener, B.; Liu, W.; Ma, L. Global Tracking of Space Debris via CPHD and Consensus. *Adv. Space Res.* **2017**, *59*, 2545–2562. [[CrossRef](#)]
43. Hill, K.; Jones, B. *TurboProp Version 4.0.*; Colorado Center for Astrodynamics Research: Boulder, CO, USA, 2009.
44. Julier, S.J.; Uhlmann, J.K. New extension of the Kalman filter to nonlinear systems. In Proceedings of the Signal Processing, Sensor Fusion, and Target Recognition VI, Orlando, FL, USA, 28 July 1997.
45. Schwarz, G. Estimating the Dimension of a Model. *Ann. Stat.* **1978**, *6*, 15–18. [[CrossRef](#)]
46. Calafiore, G.C.; Abrate, F. Distributed linear estimation over sensor networks. *Int. J. Control* **2009**, *82*, 868–882. [[CrossRef](#)]
47. Mahalanobis, P.C. *Analysis of Race-Mixture in Bengal*; Indian Science Congress: Kolkata, India, 1925; Volume 23; pp. 301–310.
48. Fantacci, C.; Vo, B.T.; Papi, F.; Vo, B.N. The Marginalized delta-GLMB Filter. *arXiv* **2015**, arXiv:1501.00926.
49. Chopra, N.; Spong, M.W. Output synchronization of nonlinear systems with relative degree one. In *Recent Advances in Learning and Control*; Springer: Berlin, Germany, 2008; pp. 51–64.
50. Li, Z.; Duan, Z.; Chen, G.; Huang, L. Consensus of multiagent systems and synchronization of complex networks: A unified viewpoint. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2010**, *57*, 213–224.
51. Xiao, F.; Wang, L. State consensus for multi-agent systems with switching topologies and time-varying delays. *Int. J. Control* **2006**, *79*, 1277–1284. [[CrossRef](#)]
52. Tsitsiklis, J.N. *Problems in Decentralized Decision Making and Computation*; Technical Report; Massachusetts Inst of Tech Cambridge Lab for Information and Decision Systems: Cambridge, MA, USA, 1984.
53. Jones, B.A.; Bryant, D.S.; Vo, B.T.; Vo, B.N. Challenges of multi-target tracking for space situational awareness. In Proceedings of the 2015 18th International Conference on Information Fusion (Fusion), Washington, DC, USA, 6–9 July 2015; pp. 1278–1285.
54. Fantacci, C. Distributed multi-object tracking over sensor networks: A random finite set approach. *arXiv* **2015**, arXiv:1508.04158.
55. DeMars, K.J.; McCabe, J.S.; Darling, J.E. Collaborative multi-sensor tracking and data fusion. In Proceedings of the 5th AAS/AIAA Space Flight Mechanics Meeting, Williamsburg, VA, USA, 11–15 January 2015.
56. Wang, B.; Yi, W.; Li, S.; Morelande, M.R. Distributed multi-target tracking via generalized multi-Bernoulli random finite sets. In Proceedings of the 2015 18th International Conference on Information Fusion (Fusion), Washington, DC, USA, 6–9 July 2015.
57. Oliva, G.; Setola, R.; Hadjicostis, C.N. Distributed Finite-Time Average-Consensus with Limited Computational and Storage Capability. *IEEE Trans. Control Netw. Syst.* **2016**, *1*, 380–391. [[CrossRef](#)]
58. Hendrickx, J.M.; Shi, G.; Johansson, K.H. Finite-Time Consensus Using Stochastic Matrices with Positive Diagonals. *IEEE Trans. Autom. Control* **2015**, *60*, 1070–1073. [[CrossRef](#)]
59. Faramondi, L.; Setola, R.; Oliva, G. Performance and robustness of discrete and finite time average consensus algorithms. *Int. J. Syst. Sci.* **2018**, *18*, 361–366. [[CrossRef](#)]
60. Sundaram, S.; Hadjicostis, C.N. Finite-Time Distributed Consensus in Graphs with Time-Invariant Topologies. In Proceedings of the 2007 American Control Conference, New York, NY, USA, 9–13 July 2007; pp. 711–716.
61. Hendrickx, J.M.; Jungers, R.; Olshevsky, A.; Vankeerberghen, G. Graph diameter, eigenvalues, and minimum-time consensus. *Automatica* **2014**, *50*, 635–640. [[CrossRef](#)]

62. Katragadda, S.; Sanmiguel, J.C.; Cavallaro, A. Consensus protocols for distributed tracking in wireless camera networks. In Proceedings of the 17th International Conference on Information Fusion (FUSION), Salamanca, Spain, 7–10 July 2014; pp. 1–8.
63. Schuhmacher, D.; Vo, B.T.; Vo, B.N. A consistent metric for performance evaluation of multi-object filters. *IEEE Trans. Signal Process.* **2008**, *56*, 3447–3457. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).