Multi-target ensemble Gaussian mixture tracking with sparse observations

Benjamin L. Reifler
The University of Texas at Austin
Sehyun Yun, Brandon A. Jones, Renato Zanetti
The University of Texas at Austin

ABSTRACT

The continuing growth of the space object population in Earth orbit will lead to increasing demand on sensor resources for space situational awareness (SSA), reducing the average frequency of observations per object, necessitating the development of multi-target tracking algorithms that can maintain custody of a large population of objects given sparse data. This paper covers work to track a simulated population of thousands of space objects, configured to create high ambiguity in data association, at multiple levels of data sparsity using a label-partitioned generalized labeled multi-Bernoulli (GLMB) filter based on an ensemble Gaussian mixture filter (EnGMF).

1. INTRODUCTION

The continuing growth of the space object population in Earth orbit will lead to increasing demand on sensor resources for space situational awareness (SSA). This will increase data sparsity, as the frequency of observations of the average space object decreases. This will especially impact sensors with small fields of view (FoVs), which provide higher quality data than full-sky sensors, but can only observe a small portion of the sky at any time. This necessitates the development of multi-target tracking (MTT) algorithms that can maintain custody of a large population of objects given sparse data. This paper covers work to track a simulated population of thousands of space objects, configured to create high ambiguity in data association, at multiple levels of data sparsity using a label-partitioned generalized labeled multi-Bernoulli (GLMB) filter based on an ensemble Gaussian mixture filter (EnGMF).

The GLMB filter is a closed-form solution to the Bayes multi-target filter recursion based on labeled random finite sets (RFSs) [1]. A labeled RFS is a set containing a random number of random vectors, each with a unique, discrete label. Labeled RFSs can represent the multi-target density of a population of distinctly identifiable objects. The GLMB RFS is a labeled RFS that maintains a table of measurement association hypotheses, referred to as components, and their associated weights. The number of components grows exponentially with each prediction and update, requiring costly truncation to remain computationally tractable.

This computational cost can be reduced by partitioning the label space to create many smaller MTT subproblems [2]. These problems can be handled in parallel, allowing the use of high performance computing facilities. Label partitioning enables tractable large-scale MTT using the GLMB filter. It also enables the use of the GLMB filter with limited-FoV sensors [3]. When too many objects are not observable at each timestep, runaway growth in the number of components can occur. This can be avoided by partitioning the label space based on which sensor's FoV an object is most likely to appear in, and those objects that are not expected to be visible simply have their tracks propagated without the full GLMB prediction and update.

Data sparsity in nonlinear MTT also necessitates accurate single-target filtering to resolve ambiguity between tracks. The EnGMF balances accuracy and computational efficiency by predicting a state probability density function (PDF) as a set of particles, then converting them to a Gaussian mixture model (GMM) via kernel density estimation (KDE) to allow measurement update using a gaussian mixture unscented Kalman filter (UKF). In this work, the EnGMF is used as the single-target filter to help resolve ambiguity due to data sparsity by increasing track prediction accuracy.

This paper presents results of applying the FoV-based label-partitioned GLMB filter to two highly ambiguous tracking scenarios in low Earth orbit (LEO): a pair of objects with consistently close spacing in the measurement space (one object following another in the same orbit) and a pair of objects with periodic conjunctions in the observation space

(two objects in similar orbits with different eccentricities). The simulated population included an additional 8000 LEO objects based on real-world two line element (TLE) data. The filter's ability to handle many different tracking subproblems was leveraged to track all 8004 objects simultaneously. Tracking performance was assessed at medium and high levels of data sparsity.

2. KERNEL-BASED ENSEMBLE GAUSSIAN MIXTURE FILTERING

The kernel-based EnGMF is designed to merge the benefits of particle filter (PF) and Gaussian sum filter (GSF). In the EnGMF, KDE, which is the process of estimating the unknown PDF of a random variable [4], is used to integrate a PF with a GSF; that is, each propagated particle is approximated as a Gaussian component with a covariance matrix estimated by bandwidth selection for KDE [5, 6, 7].

As a recursive algorithm, the prior distribution $p(x_{k-1}|y_{k-1})$ is approximated by N independent and identically distributed (i.i.d.) particles $x_{k-1}^{(i)}$ such that

$$p(x_{k-1}|y_{k-1}) \approx \sum_{i=1}^{N} \frac{1}{N} \delta(x_{k-1} - x_{k-1}^{(i)}), \tag{1}$$

where k is an integer that indicates the discrete time step, y is a measurement vector, and $\delta(\cdot)$ is the Dirac delta function. As in the bootstrap particle filter (BPF) [8], a set of particles at the next time step is obtained using the Markov transition kernel $p(x_k|x_{k-1})$. The Markov kernel refers to the dynamics of a system.

The propagated particles are converted into Gaussian mixtures using KDE in the next step. The approximated GMM is then expressed as follows:

$$p(x_k) \approx \sum_{i=1}^{N} \frac{1}{N} n(x_k; x_{k|k-1}^{(i)}, B), \tag{2}$$

where the bandwidth matrix B is can be estimated by [3]

$$B = \beta \hat{P}, \tag{3}$$

where β is the bandwidth parameter, $0 \leq \beta \leq 1$, and $\hat{P}$ is the sample covariance matrix calculated from the propagated particles. The mean of the i^{th} Gaussian component is the particle $x_{k|k-1}^{(i)}$ and all GMM weights are equal to each other, $1/N$. The covariance matrix of the GMM is determined by the bandwidth parameter. In this paper, to reduce the KDE computational burden, Silverman's rule of thumb [9] is used to estimate the bandwidth (i.e., covariance) matrix B_S as follows [7]:

$$B_S = \beta_S \hat{P} = \left(\frac{4}{n_x + 2} \right)^{\frac{2}{n_x+4}} N^{-\frac{2}{n_x+4}} \hat{P}, \tag{4}$$

where n_x is the dimension of the state. Note that the optimal mean integrated squared error (MISE) bandwidth parameter is obtained with Silverman's rule of thumb when the sampling distribution is Gaussian [9]. Even though the sampling distribution is not close to Gaussian, it will result in conservative (large) estimates, which is a desirable feature since inaccuracies leads to conservatism rather than divergence [7].

Finally, the measurement information is incorporated by the measurement update of the GSF. N i.i.d. samples are then drawn from the GMM approximation of the posterior distribution for the next iteration.

3. FOV-BASED LABEL-PARTITIONED GLMB FILTERING

3.1 Notation

The following notation will be used in this section. Lower- and upper-case letters represent single- and multi-object variables (i.e., vectors and sets of vectors), respectively, while bold symbols represent labeled states and their distributions. Blackboard bold letters represent spaces. To simplify notation, the subscript k for variables at time k

will be omitted, and variables at the following time $k+1$ will be denoted with a subscript plus sign. The multi-object exponential is defined as

$$f^X = \prod_{x \in X} f(x)\,, \tag{5}$$

the Kronecker delta is defined as

$$\delta_Y(X) = \begin{cases} 1, & X = Y\,, \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

and the inclusion function is defined as

$$1_Y(X) = \begin{cases} 1, & X \subseteq Y\,, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

The function $\mathcal{F}$ denotes the set of all finite subsets of a space. Given a state space $\mathbb{X}$, a random finite set (RFS) is a random variable on $\mathcal{F}(\mathbb{X})$. A labeled RFS is created by augmenting each element of an RFS with a discrete label, resulting in a random variable on $\mathcal{F}(\mathbb{X}\times\mathbb{L})$, where $\mathbb{L}$ is the label space. For example, a realization of a labeled RFS with n elements is

$$\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} = \{(x_1, l_1), \ldots, (x_n, l_n)\} \subset \mathbb{X}\times\mathbb{L}\,. \tag{8}$$

All realizations of a labeled RFS must contain only distinct labels. The distinct label indicator is defined as

$$\Delta(\boldsymbol{X}) = \delta_{|\boldsymbol{X}|}(|\mathcal{L}(\boldsymbol{X})|)\,, \tag{9}$$

where $\mathcal{L}((x,l)) = l$ denotes the projection of the label-augmented state space $\mathbb{X}\times\mathbb{L}$ onto its discrete label space $\mathbb{L}$.

3.2 Labeled Random Finite Sets

The filter used in this paper is built using two types of RFSs: δ-generalized labeled multi-Bernoulli (δ-GLMB) and labeled multi-Bernoulli (LMB) RFSs.

3.2.1 Generalized Labeled Multi-Bernoulli RFSs

A GLMB RFS has a PDF of the form

$$\boldsymbol{\pi}(\boldsymbol{X}) = \Delta(\boldsymbol{X}) \sum_{\xi \in \Xi} w^{(\xi)}(\mathcal{L}(\boldsymbol{X})) \left(p^{(\xi)}\right)^{\boldsymbol{X}}\,, \tag{10}$$

where Ξ is a discrete space, each $p^{(\xi)}$ is a PDF, and each weight $w^{(\xi)}$ is non-negative with

$$\sum_{(I,\xi) \in \mathcal{F}(\mathbb{L})\times\Xi} w^{(\xi)}(I) = 1\,. \tag{11}$$

Note that the PDFs $p^{(\xi)}$ are independent. The GLMB filter used in this work is based on an alternative formulation known as a δ-GLMB, which may be parameterized by a set of components $(I,\xi) \in \mathcal{F}(\mathbb{L})\times\Xi$, with associated weights $w^{(I,\xi)} = w^{(\xi)}(I)$. In the MTT context, these components are data association hypotheses, with I being the set of objects that exist, ξ being the combined measurement association history of those objects, and $w^{(I,\xi)}$ being the probability that the hypothesis is true. The PDF of a δ-GLMB RFS is

$$\boldsymbol{\pi}(\boldsymbol{X}) = \Delta(\boldsymbol{X}) \sum_{(I,\xi) \in \mathcal{F}(\mathbb{L})\times\Xi} w^{(I,\xi)} \delta_I(\mathcal{L}(\boldsymbol{X})) \left(p^{(\xi)}\right)^{\boldsymbol{X}}\,, \tag{12}$$

where $p^{(\xi)}(x,l)$ is the state-space PDF of x given the association history ξ and label l. This can be equivalently expressed as

$$\boldsymbol{\pi}(\boldsymbol{X}) = \Delta(\boldsymbol{X})\, 1_{\mathcal{F}(\mathbb{L})}(\mathcal{L}(\boldsymbol{X})) \sum_{\xi \in \Xi} w^{(\mathcal{L}(\boldsymbol{X}),\xi)} \left(p^{(\xi)}\right)^{\boldsymbol{X}}\,, \tag{13}$$

where $\Delta(\boldsymbol{X})$ ensures that only RFSs with distinct labels are possible, $1_{\mathcal{F}(\mathbb{L})}(\mathcal{L}(\boldsymbol{X}))$ ensures that only RFSs that contain a finite number of labels, all of which belong to the current label space, are possible, and the final sum represents the state-space multi-target density over all hypotheses (I,ξ) where $I = \mathcal{L}(\boldsymbol{X})$.

3.2.2 Labeled Multi-Bernoulli RFSs

An LMB RFS is similar to a GLMB RFS, but instead of maintaining multiple data association hypotheses, it collapses them into a single hypothesis. An LMB may be parameterized by $\left\{\left(r^{(l)}, p^{(l)}\right)\right\}_{l\in\mathbb{L}}$, where $r^{(l)}$ is the probability that object l exists and $p^{(l)}$ is its state-space PDF. The PDF of an LMB RFS has the form

$$\boldsymbol{\pi}(\boldsymbol{X}) = \Delta(\boldsymbol{X})\, w(\mathcal{L}(\boldsymbol{X}))\, p^{\boldsymbol{X}}\,, \tag{14}$$

where

$$w(I) = \prod_{l\in\mathbb{L}}\left(1-r^{(l)}\right)\prod_{l\in\boldsymbol{I}}\frac{1_{\mathbb{L}}(l)\, r^{(l)}}{1-r^{(l)}}\,, \tag{15}$$

$$p(x,l) = p^{(l)}(x)\,. \tag{16}$$

Labeled multi-Bernoulli RFSs are used in this GLMB filter to model spontaneous birth, but the scenario presented here used a static population of objects and did not include birth.

3.3 Joint Prediction and Update

The number of components in a GLMB tends to grow exponentially during each prediction and measurement update, so components with relatively low weights are truncated to keep the filter tractable while minimizing the L_1 truncation error [10]. In the classic form of the GLMB filter (see [1]), two hypothesis truncation operations are required: one after prediction and another after the measurement update. As presented in [10], a mathematically equivalent formulation of the filter yields a single hypothesis truncation operation. This joint prediction and update formulation is used here because it is less computationally costly.

In this form of the filter, given the filtering density in Eq. (12), the filtering density at the following time is given by

$$\boldsymbol{\pi}_+(\boldsymbol{X}_+) \propto \Delta(\boldsymbol{X}_+)\sum_{I,\xi,I_+,\vartheta_+} w^{(I,\xi)} w^{(I,\xi,I_+,\vartheta_+)}(Z_+)\,\delta_{I_+}(\mathcal{L}(\boldsymbol{X}_+))\left(p_+^{(\xi,\vartheta_+)}(\cdot\mid Z_+)\right)^{\boldsymbol{X}_+}\,, \tag{17}$$

where $I\in\mathcal{F}(\mathbb{L})$, $\xi\in\Xi$, $I_+\in\mathcal{F}(\mathbb{L}_+)$, and $\vartheta_+\in\Theta_+$, where $\mathbb{L}_+ = \mathbb{L}\cup\mathbb{B}_+$, $\mathbb{B}_+$ is the space of object labels that could be born at this time, Θ_+ is the set of positive one-to-one maps $\vartheta_+ : \mathbb{L}_+ \to \{0:|Z_+|\}$ assigning measurements in the current measurement set Z_+ to labels in the GLMB ($\vartheta_+(l) = 0$ implies that label l is not assigned a measurement). Additionally,

$$w^{(I,\xi,I_+,\vartheta_+)}(Z_+) = \left(1-\bar{p}_S^{(\xi)}\right)^{I-I_+}\left(\bar{p}_S^{(\xi)}\right)^{I\cap I_+}\left(1-r_{B,+}\right)^{\mathbb{B}_+-I_+}\left(r_{B,+}\right)^{\mathbb{B}_+\cap I_+}\left(\bar{\psi}_+^{(\xi,\vartheta_+)}(\cdot\mid Z_+)\right)^{I_+}\,, \tag{18}$$

$$\bar{p}_S^{(\xi)}(l) = \left\langle p_S(\cdot,l)\,, p^{(\xi)}(\cdot,l)\right\rangle\,, \tag{19}$$

$$\bar{\psi}_+^{(\xi,\vartheta_+)}(l_+\mid Z_+) = \left\langle \bar{p}_+^{(\xi)}(\cdot,l_+)\,, \psi_+^{(\vartheta_+(l_+))}(\cdot,l_+\mid Z_+)\right\rangle\,, \tag{20}$$

$$p_+^{(\xi,\vartheta_+)}(x_+,l_+\mid Z_+) = \frac{\bar{p}_+^{(\xi)}(x_+,l_+)\,\psi_+^{(\vartheta_+(l))}(x_+,l_+\mid Z_+)}{\bar{\psi}_+^{(\xi,\vartheta_+)}(l_+\mid Z_+)}\,, \tag{21}$$

$$\bar{p}_+^{(\xi)}(x_+,l_+) = 1_{\mathbb{B}_+}(l_+)\, p_{B,+}(x_+,l_+) + 1_{\mathbb{L}}(l_+)\,\frac{\left\langle p_S(\cdot,l_+)\, f_{S,+}(x_+\mid\cdot,l_+)\,, p^{(\xi)}(\cdot,l_+)\right\rangle}{\bar{p}_S^{(\xi)}(l_+)}\,, \tag{22}$$

$$\psi_+^{(i)}(x_+,l_+\mid Z_+) = \delta_0(i)\,(1-p_D(x_+,l_+)) + (1-\delta_0(i))\,\frac{p_D(x_+,l_+)\, g\left(z_{+,i}\,\middle|\, x_+,l_+\right)}{\kappa\left(z_{+,i}\right)}\,, \tag{23}$$

where $r_{B,+}(l_+)$ is the probability that object l_+ is born, $p_{B,+}(x_+,l_+)$ is the single-target probability density for the newborn object, $p_S(x_+,l_+)$ is the probability of survival from one step to the next, $p_D(x_+,l_+)$ is the probability of detection, $f_{S,+}(x_+\mid\cdot,l_+)$ is the surviving object transition density, $g\left(z_{+,\vartheta_+(l_+)}\,\middle|\, x_+,l_+\right)$ is the measurement likelihood, and $\kappa(z)$ is the expected clutter intensity for a given measurement z.

3.3.1 Hypothesis Truncation

The most computationally intensive step of the GLMB filter is the truncation of considered association hypotheses [2]. Measurement gating is a simple way to significantly reduce the number of associations considered. Gating is performed by calculating the Mahalanobis distance

$$d_M(z) = \sqrt{(z-\mu)^\mathsf{T} \Sigma^{-1} (z-\mu)} \tag{24}$$

between a measurement and an object's PDF in measurement space and eliminating associations that exceed some distance threshold.

Another way to eliminate unlikely hypotheses from consideration is through the use of ranked assignment algorithms, such as Murty's algorithm, which look at the cost of each individual association. The computational cost of processing the i^{th} component with Murty's algorithm is $O\left(C_i (N_i + M)^3\right)$, where C_i is the number of allocated posterior components, N_i is the number of unique target labels, and M is the number of measurements. An alternative algorithm when considering many data association and birth/death hypotheses at a single time is based on Gibbs sampling, which has a computational cost for the i^{th} component of $O\left(C_i N_i^2 M\right)$ [2]. The Gibbs sampler is a computationally efficient special case of the Metropolis–Hastings algorithm, and by properly defining the probability distribution to sample, it can be used to quickly find a number of relatively highly-weighted associations [10]. Despite the Gibbs sampling algorithm's better scalability, Murty's algorithm is generally better for updating GLMBs containing a low number of possible objects, because this results in a relatively small number of viable data association hypotheses.

For a GLMB containing P object labels, K of which are from the birth model, the cost of each assignment j, where $j = 0$ indicates no assignment and $j = -1$ indicates that the target does not exist, is

$$\eta_i^{(I,\xi)}(j) = \begin{cases} 1 - r_{B,+}(l_i), & 1 \le i \le K,\ j < 0, \\ r_{B,+}(l_i)\, \bar{\psi}_+^{(\xi,j)}(l_i \mid Z_+), & 1 \le i \le K,\ j \ge 0, \\ 1 - \bar{p}_S^{(\xi)}(l_i), & K+1 \le i \le P,\ j < 0, \\ \bar{p}_S^{(\xi)}(l_i)\, \bar{\psi}_+^{(\xi,j)}(l_i \mid Z_+), & K+1 \le i \le P,\ j \ge 0. \end{cases} \tag{25}$$

3.3.2 Implementation

The GLMB filter is implemented by storing a table of tracks, each of which comprises a label and state-space PDF. Each track's PDF incorporates its association history. Each component is parameterized as a set of track indices. To predict and update, all existing tracks are predicted, new tracks are added from the LMB birth model, and all tracks are updated according to each possible measurement association. Then, a cost matrix is calculated for each prior component using Eq. (25) and ranked assignment is used to decide which components are included in the posterior GLMB. The weight of each posterior component is a function of its cost. This method can generate multiple instances of the same component, so all duplicate components are merged and their weights are added together. Finally, components with low weights are pruned, the lowest weighted components are truncated to meet the upper limit on the number of components, and unused tracks are deleted.

3.4 Label Space Partitioning by Sensor Field of View

Partitioning the label space before performing the joint prediction and update can significantly reduce the computational load of tracking large numbers of objects [2]. Here, the same idea of decomposing the MTT problem by partitioning the label space is used to enable the use of the GLMB filter with limited-FoV sensors in the context of space object tracking. The partition is constructed by separating the state space into regions corresponding to the FoV of each sensor and grouping the labels of the objects that are expected to be in each FoV. By assuming that objects outside of the FoV of any sensor cannot be detected and cannot die, the set of labels outside of any sensor's FoV does not need to be merged into one large GLMB, keeping the size of each group small. Additionally, because each group's prediction and update are entirely independent, the resulting partitioned GLMB filter is highly parallelizable.

To update the partition before prediction and update, the objects near the FoV of each sensor are identified. We define an object as being near the FoV if the mean of its predicted PDF given the highest-weighted hypothesis is within some distance of the sensor limits after being mapped into the measurement space. Note that while an object may have its

mean near the FoV of one sensor, a significant portion of its predicted PDF may be outside of it. This is accounted for by calculating p_V, the probability that the object is in the FoV, which will be discussed in the following implementation section.

Groups in the prior partition may end up with labels in multiple regions after prediction. When this occurs, these groups are split by marginalizing over the labels in each region, which has been shown to minimize the Kullback–Leibler divergence from the orignal GLMB [2]. Given a GLMB density of the form given in Eq. (10) and a partition $G = \left\{\mathbb{L}^{(1)}, \ldots, \mathbb{L}^{(N)}\right\}$ of N groups, the n^{th} marginalized density is given by

$$\boldsymbol{\pi}^{(n)}(\boldsymbol{X}) = \Delta(\boldsymbol{X}) \sum_{\xi \in \Xi} w^{(\xi,n)}(\mathcal{L}(\boldsymbol{X})) \left(p^{(\xi)}\right)^{\boldsymbol{X}}, \tag{26}$$

where

$$w^{(\xi,n)}(I) = \sum_{J \in \mathcal{F}(\mathbb{L}) - \mathcal{F}\left(\mathbb{L}^{(n)}\right)} w^{(\xi)}(I \cup J) \,. \tag{27}$$

Then, all groups in each sensor's FoV are merged into one larger GLMB using the GLMB product operation:

$$\boldsymbol{\pi}(\boldsymbol{X}) \approx \prod_{n=1}^{N} \boldsymbol{\pi}^{(n)}(\boldsymbol{X}) \,. \tag{28}$$

Fig. 1 shows how the partition update fits into the GLMB joint predict–update step.

3.4.1 Implementation

The GLMB product shown in Eq. (28) can exponentially increase the number of components, which may exceed the available memory. To avoid this, the intermediate GLMB is pruned and truncated after each product operation:

$$\boldsymbol{\pi}(\boldsymbol{X}) \approx \text{prune\_truncate}\Big(\ldots \text{prune\_truncate}\Big(\boldsymbol{\pi}^{(1)}(\boldsymbol{X})\, \boldsymbol{\pi}^{(2)}(\boldsymbol{X})\Big) \ldots \boldsymbol{\pi}^{(N)}(\boldsymbol{X})\Big) \,. \tag{29}$$

In the filter, the possibility that an object with its mean near one sensor's FoV may have a track with a significant portion of its probability density outside of that FoV is accounted for by calculating p_V by repeated random sampling of its predicted PDF:

$$p_V(x_+, l_+) = \frac{1}{N_V} \sum_{n=1}^{N_V} 1_V\left(h\left(x_+^{(n)}\right)\right), \tag{30}$$

where

$$x_+^{(n)} \sim p_+^{(\xi)}(\cdot, l_+) \,, \tag{31}$$

$V \subseteq \mathbb{Z}$ is the sensor's FoV, and N_V is the number of samples. This probability is then used to apply the assumptions stated above: only objects in the FoV can be detected, so p_D becomes $p_V p_D$; only objects in the FoV can be born, so $r_{B,+}$ becomes $p_V r_{B,+}$; and objects outside of the FoV cannot die, so p_S becomes $1 - p_V\,(1 - p_S)$.

4. SCENARIO DESCRIPTION

To assess the filter's ability to simultaneously handle many different space object tracking subproblems, simulated measurements of thousands of objects were generated from a set of eight terrestrial sensors.

4.1 Measurements and Sensor Tasking

The sensor locations are shown in Table 1. Each simulated sensor produced azimuth, elevation, and range measurements, and all except for Sensor 7 also produced range-rate measurements, with measurement noise covariance

$$R = \text{diag}\left(\sigma_\vartheta^2, \sigma_\vartheta^2, \sigma_\rho^2, \sigma_{\dot\rho}^2\right), \tag{32}$$

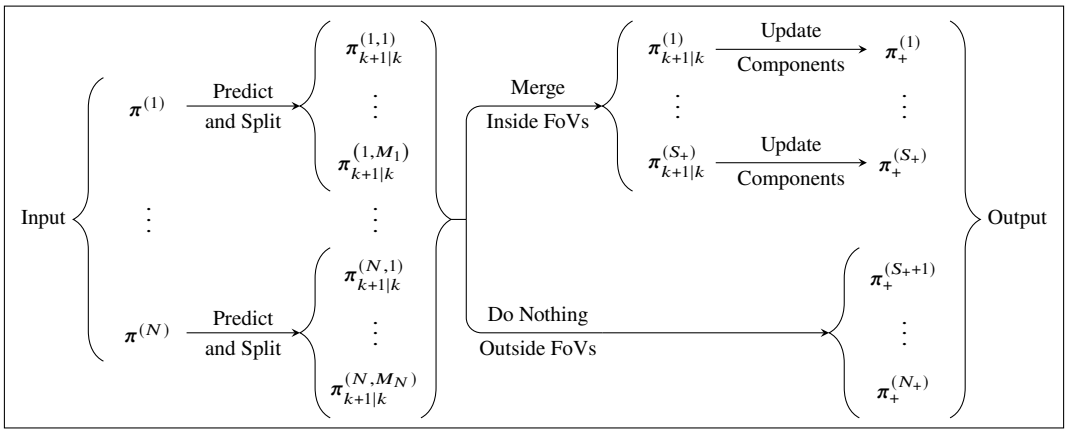

Fig. 1: Diagram of the FoV-based label-partitioned GLMB joint prediction and update with N prior groups, N_+ posterior groups, and S_+ active sensors.

Table 1: Sensor locations.

#	Latitude (deg)	Longitude (deg)	Altitude (m)
1	−7.907	−14.403	58.328
2	39.136	−121.351	142.520
3	41.753	−70.538	109.992
4	64.300	−149.191	164.009
5	30.572	−86.215	73.640
6	54.362	−0.671	211.195
7	9.395	167.479	−23.619
8	42.617	−71.491	156.466

where $\sigma_\vartheta = 10\,\text{arcsec}$, $\sigma_\rho = 10\,\text{m}$, $\sigma_{\dot\rho} = 10\,\text{cm/s}$, and constant probability of detection $P_D = 0.95$. Each sensor had a circular FoV with a radius of 2° and the radius for partitioning was 2.5°. Clutter was not modeled. The clutter density in the filter was set to a constant $\kappa = 10^{-10}$, rather than zero, for two reasons: to avoid numerical issues caused by Eq. (23) and to add robustness in the unlikely event that an object was grouped incorrectly in the partition, leading to more measurements than the expected number of labels in sensor's FoV.

Sensor tasking was performed offline based on the true states of the objects. Each sensor was tasked to point at the object in its field of regard with the longest time since it was last in the FoV of any sensor, subject to the constraint that no object could be within 1° of multiple sensors' FoVs, to avoid the case where one object could end up in multiple groups within the partition.

4.2 Object Population

The simulated objects comprised two pairs of satellites designed to produce high ambiguity in data association and 8000 LEO objects based on real-world TLE data. The initial orbital elements of the four ambiguous satellites are shown in Table 2. The first pair (Satellites 1 and 2) had a near-constant separation of approximately 240 km and the second pair (Satellites 3 and 4) had periodic conjunctions in the measurement space. The configuration of the satellites in these pairs is illustrated in Fig. 2. The distribution of orbital elements of the other 8000 objects is shown in Fig. 3.

The objects' orbits were propagated with perturbations due to the Sun and Moon's gravity and a 16×16 spherical harmonic expansion model of the Earth's gravitational field. The simulation timestep was $\Delta t = 10\,\text{s}$. Drag was not included because some objects were in very low orbits and orbital decay would cause numerical issues. The filter used the same orbital dynamics model as was used to propagate the objects' true states and no process noise was included. Using the true dynamics was justified because initial error and measurement noise created significant uncertainty over time. However, as discussed in section 6, reducing propagation accuracy and adding process noise are two of the first things we will do in continuing this work.

Table 2: Initial osculating Keplerian elements for the four ambiguous objects.

#	Semi-maj. axis (km)	Ecc.	Incl. (deg)	RAAN (deg)	Arg. Periapsis (deg)	Mean Anomaly (deg)
1	7000	0.001	54	359.821	359.863	0.000
2	7000	0.001	54	359.821	359.863	−0.002
3	7000	0.001	54	359.821	359.863	19.961
4	7000	0.002	54	359.821	359.863	19.922

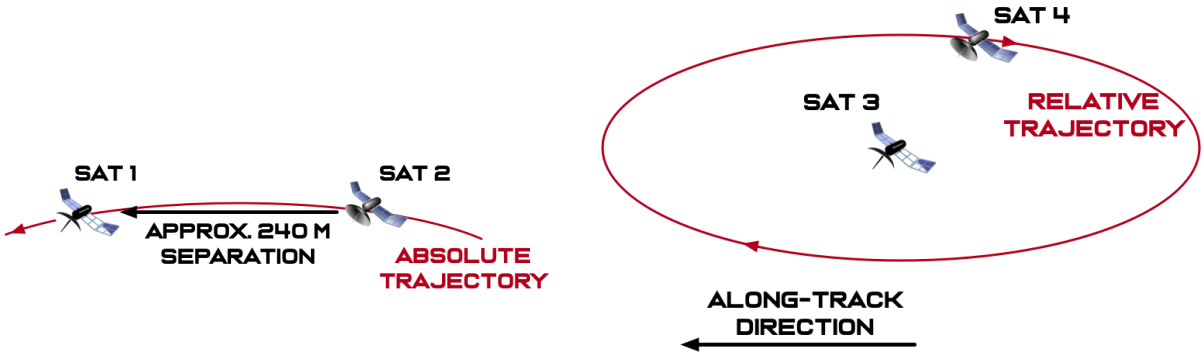

Fig. 2: *Left:* Satellites 1 and 2 with fixed separation. *Right:* Satellites 3 and 4 with periodic conjunctions.

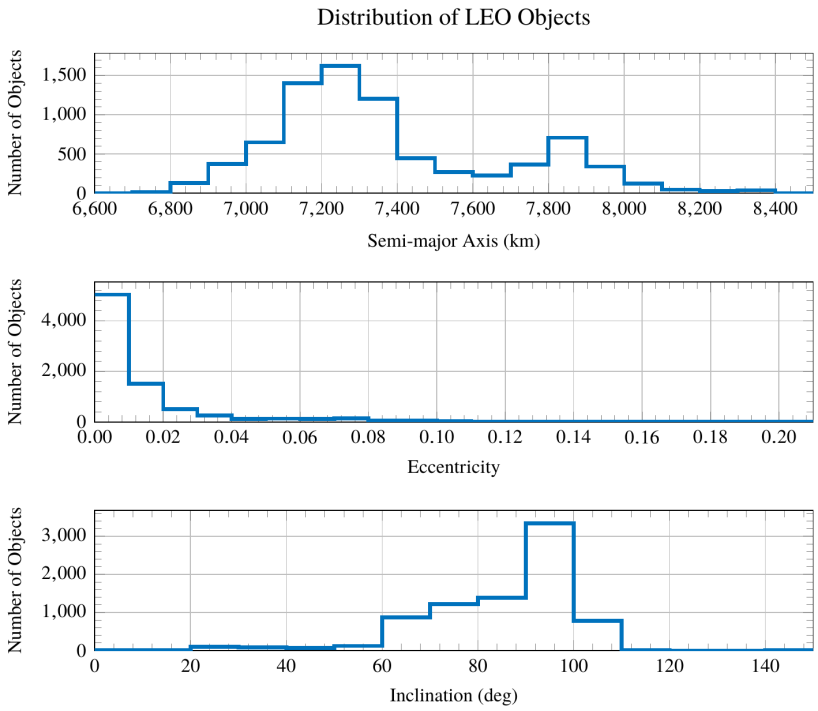

Fig. 3: Distribution of orbital elements of the 8000 LEO objects.

4.3 Initialization and Filter Parameters

The filter was initialized with one track per object, each in a separate group of the initial partition. Each track's PDF was initialized by single-target filtering via the (single-Gaussian) UKF for 150 simulation steps with increased measurement noise covariance $100^2 R$, timestep $\Delta t = 5$ min, and initial standard deviations 1 km and 10 m/s along each axis for position and velocity, respectively. No tracks were allowed to die (i.e., $P_S = 1$) and there was no birth (i.e., $\mathbb{B} = \emptyset$). In the multi-target filter, the maximum total number of components and the maximum total number of new components after each prediction and update were both set to 10^5, the pruning threshold was 10^{-14}, and the gating threshold was 10 standard deviations. In the single-target filter, 500 particles were used to parameterize the state PDF.

5. RESULTS

The filter was tested in two data sparsity test cases. In the medium sparsity case, measurements were available at each simulation step (every 10 s), which resulted in an average of one observation per object every 1.5 h. The high sparsity case used the same initial conditions and sensor tasking, but measurements were only used every sixth step (1 min), resulting in an average of one observation per object every 9.3 h. The medium sparsity simulation had a duration of two simulated days and the high sparsity simulation covered five days.

5.1 Medium Sparsity

As shown in Figs. 4 and 5, the filter reached steady state in both the overall root mean square (RMS) tracking error and GLMB complexity by the end of the second day. In Fig. 5, the bottom subplot shows that the final number of groups was roughly half the number of objects, meaning the average group contained two objects. The top subplot shows that the total number of components over all groups of the partition reached about 20,000, indicating that the average group had four or five data association hypotheses. Finally, the middle subplot shows that the number of tracks also reached about 20,000, meaning that the average object had two or three tracks at any given time. The average computation time for each simulation step was proportional to the number of tracks, beginning at about 2.5 s and ending around 6 s, likely due to the low number of objects per component (making component update faster) and the computational cost of EnGMF prediction.

Figs. 6 and 7 show that the objects' individual tracking errors were well-bounded by the estimated uncertainty, and that the final error for all objects was under 1 km. Additionally, Fig. 7 shows that the final estimation errors and uncertainties in tracking the four ambiguous satellites was lower than the distance between the satellites in each pair, meaning that the filter was able to differentiate between the objects in each pair.

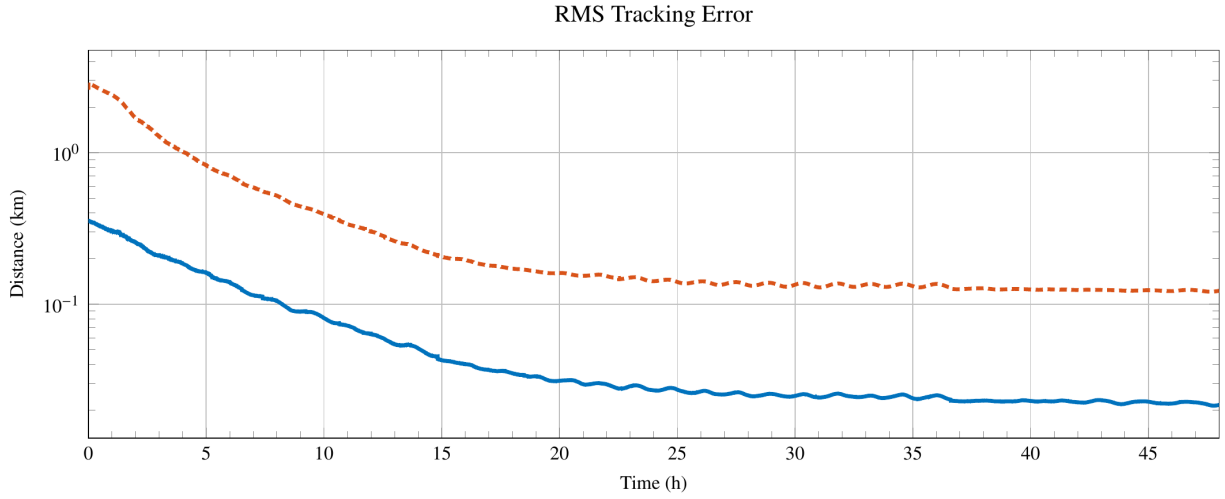

Fig. 4: Overall tracking error (solid) and uncertainty (dashed) for the medium sparsity case.

5.2 High Sparsity

Unlike in the medium sparsity case, Figs. 8 and 9 show that the filter did not reach steady state by the end of this simulation. However, Fig. 10 shows that the maximum error was decreasing and well bounded by its estimated

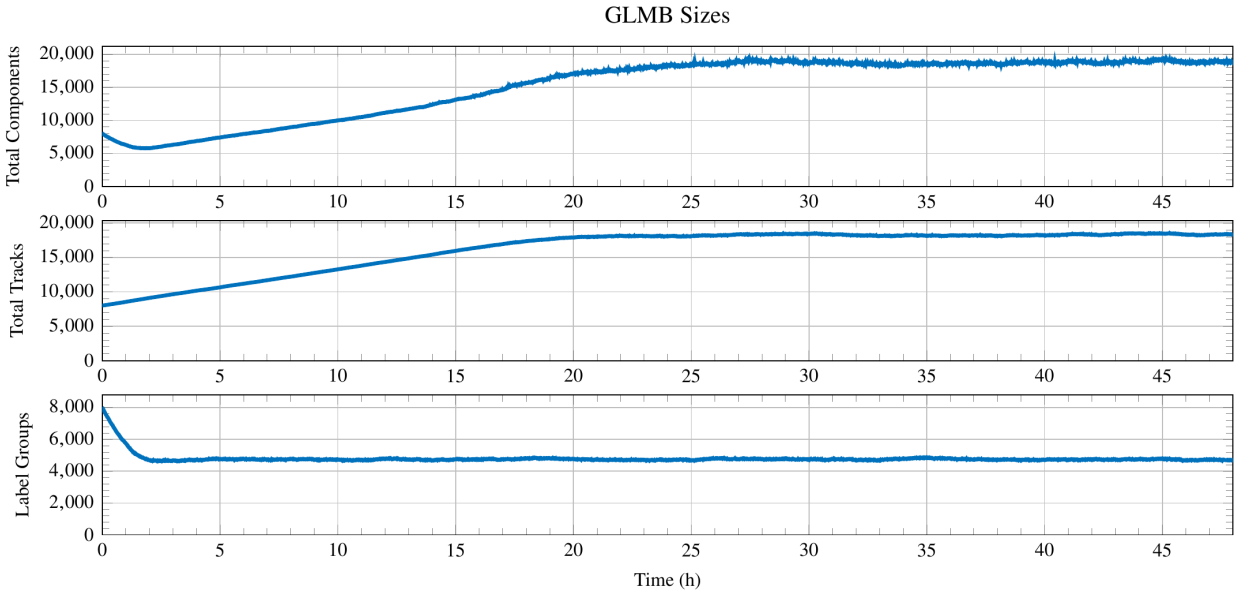

Fig. 5: Label partitioned GLMB complexity over time for the medium sparsity case.

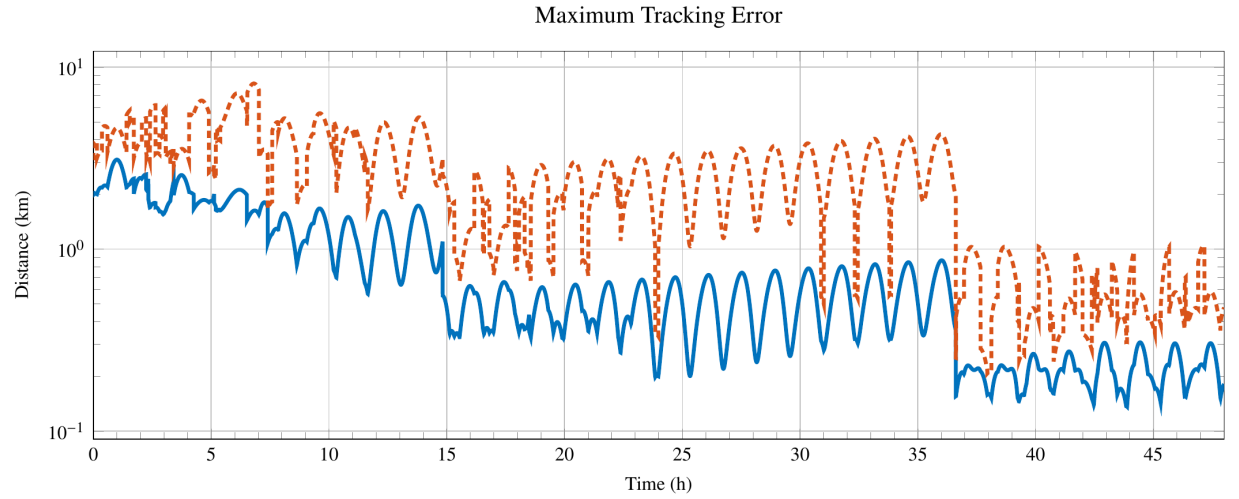

Fig. 6: Maximum tracking error (solid) and that object's estimated uncertainty (dashed) at each step for the medium sparsity case.

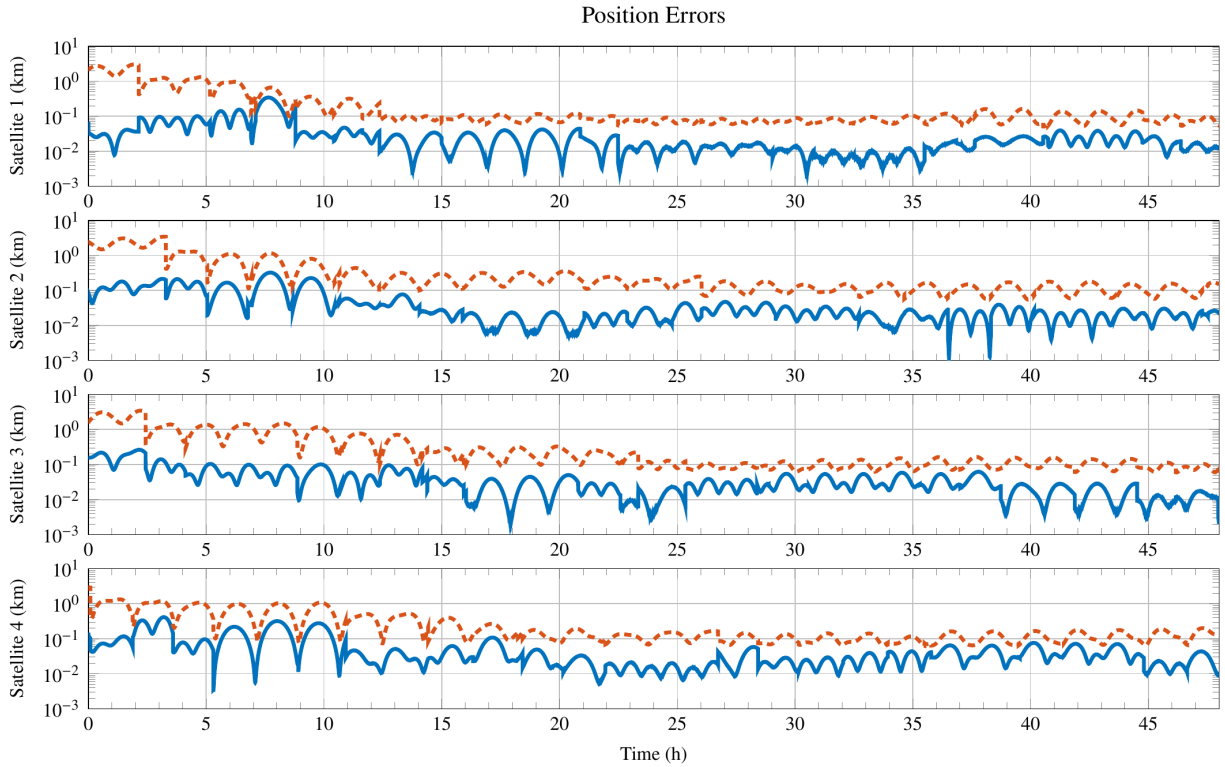

Fig. 7: Tracking errors (solid) and uncertainties (dashed) for the ambiguous object pairs for the medium sparsity case.

uncertainty, indicating that the filter did not and likely would not lose custody of any objects if the duration were increased. Additionally, the final maximum error was again below 1 km. As in the medium sparsity case, Fig. 11 shows that the filter was ultimately able to differentiate between the satellites in each ambiguous pair.

6. CONCLUSION AND FUTURE WORK

The combination of FoV-based label-partitioned GLMB filtering and kernel-based ensemble Gaussian mixture filtering has been applied to simulated measurements of over 8000 LEO objects, including two pairs of satellites in formations designed to result in high ambiguity in data assocation, at multiple levels of data sparsity. The filter was able to maintain custody of all objects without requiring an unreasonable number of tracks or components. The filter was also able to differentiate between the objects in each pair, even at high sparsity.

However, this scenario may have been too easy for the filter due to low measurement noise and exact knowledge of the dynamics. In continuing this work, we will increase measurement noise significantly and introduce more state error and uncertainty over time by reducing the accuracy of the filter's propagation model and adding process noise to compensate. This will increase ambiguity in data association, especially for the two satellite pairs, which may require a different approach to remain tractable. This may draw on the adaptive LMB filter [11], which combines the accuracy of the GLMB filter with the computational efficiency of the LMB filter, its single hypothesis approximation. Other improvements we will make to these test cases include approximately doubling the number of objects and adding more complicated scenarios including maneuvering targets.

7. ACKNOWLEDGMENT

This work was sponsored in part by DARPA under STTR contract number 140D0420C0062.

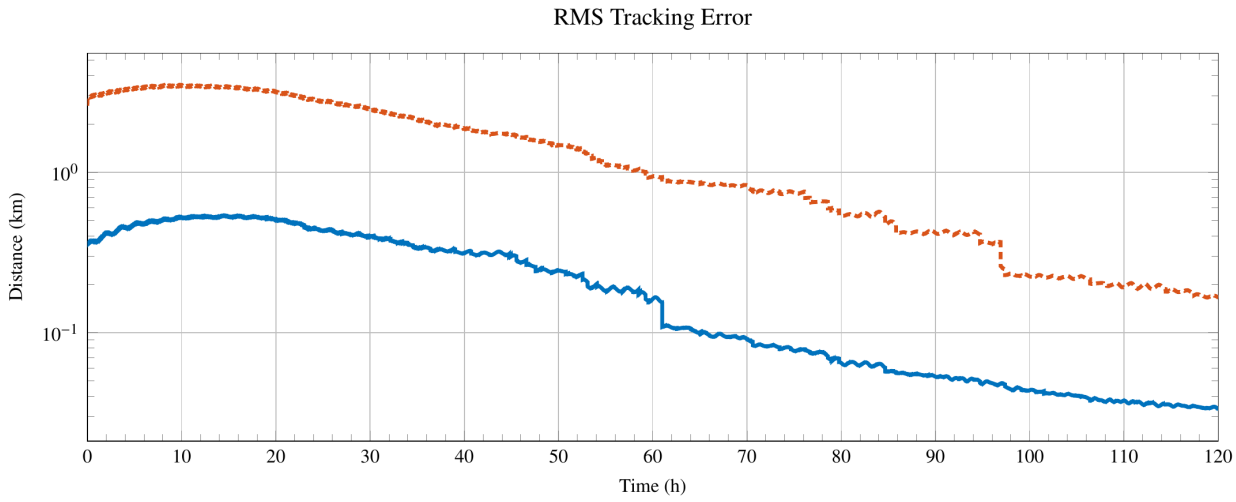

Fig. 8: Overall tracking error (solid) and uncertainty (dashed) for the high sparsity case.

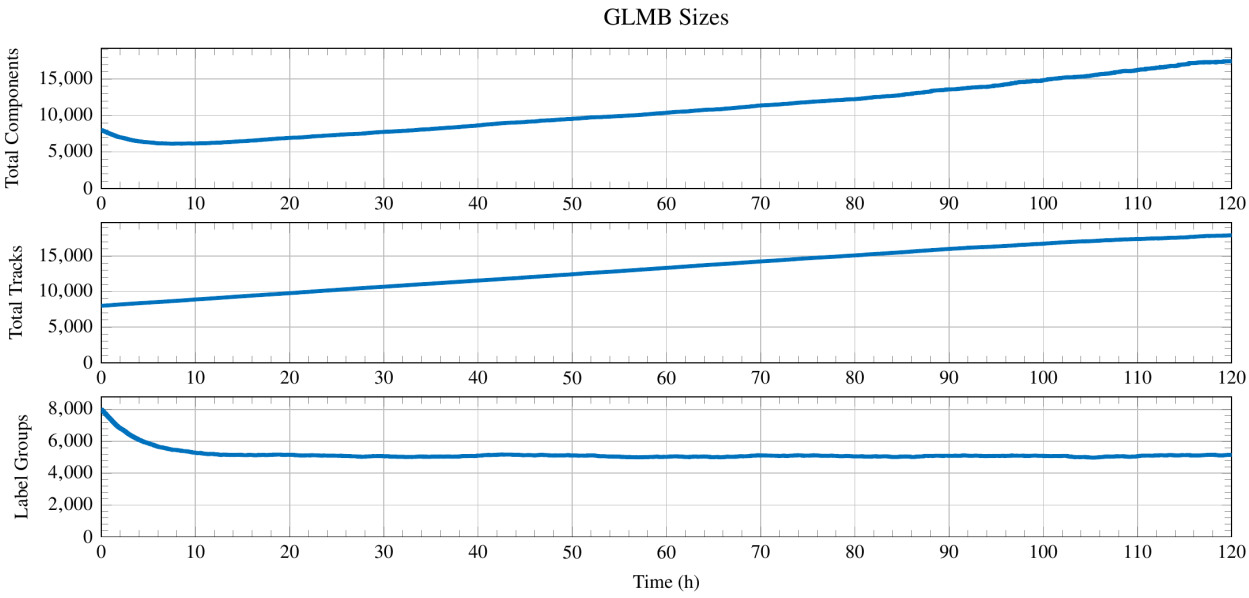

Fig. 9: Label partitioned GLMB complexity over time for the high sparsity case.

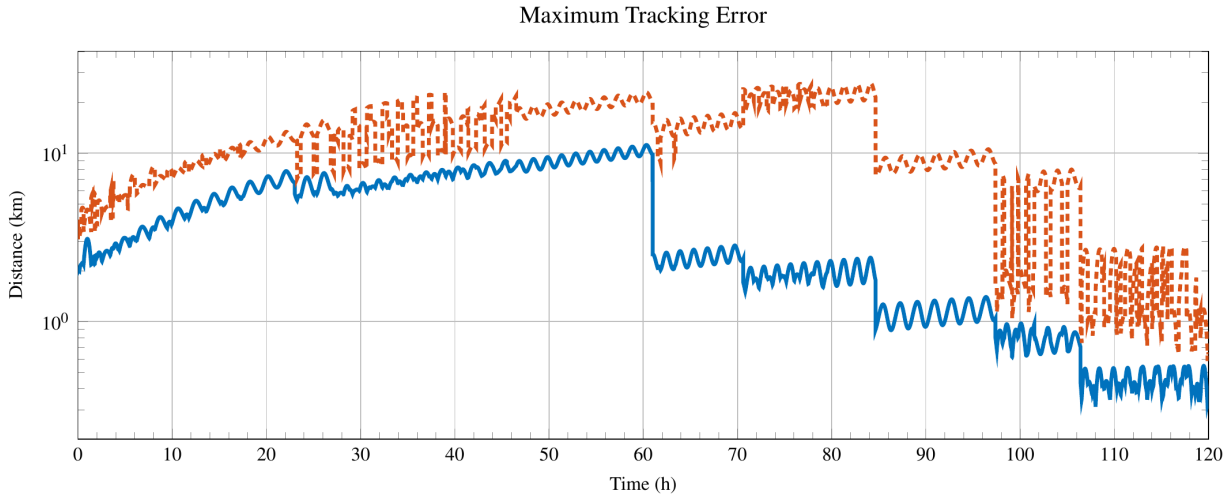

Fig. 10: Maximum tracking error (solid) and that object's estimated uncertainty (dashed) at each step for the high sparsity case.

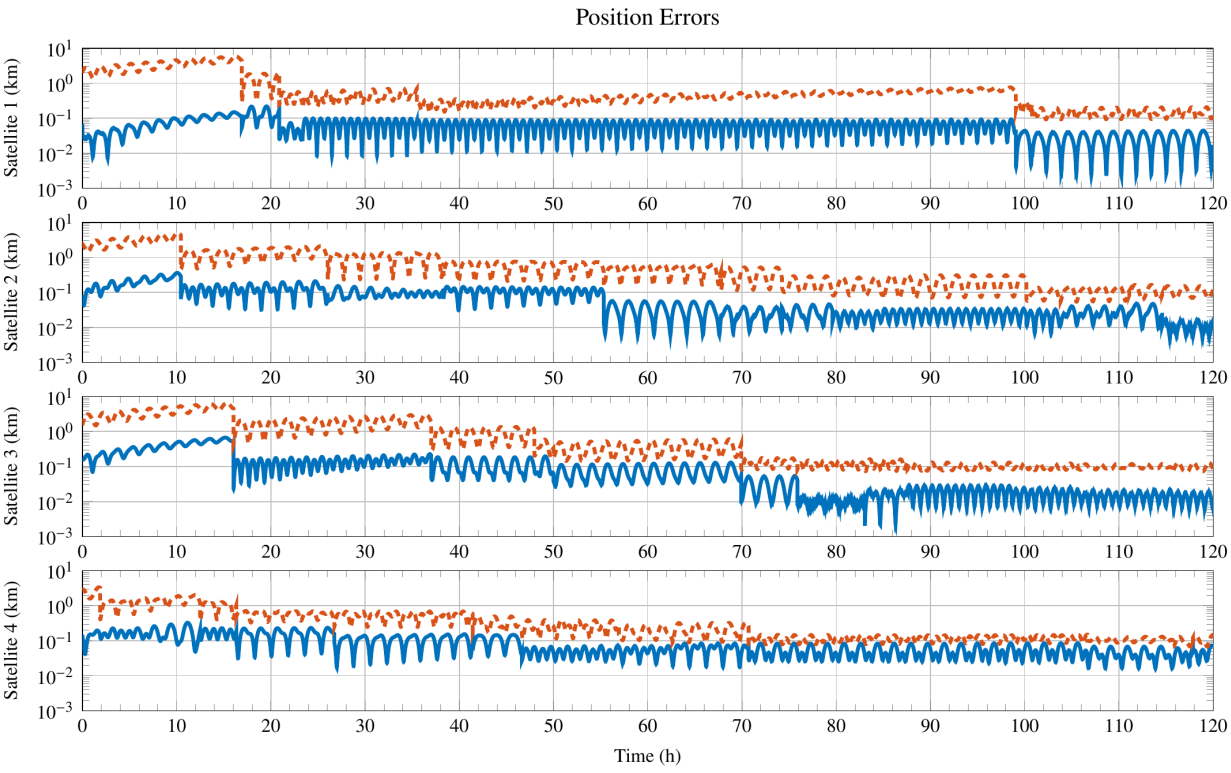

Fig. 11: Tracking errors (solid) and uncertainties (dashed) for the ambiguous object pairs for the high sparsity case.

8. REFERENCES

[1] Ba-Tuong Vo and Ba-Ngu Vo. Labeled random finite sets and multi-object conjugate priors. *IEEE Transactions on Signal Processing*, 61(13), July 2013. doi:10.1109/TSP.2013.2259822.

[2] Michael Beard, Ba-Tuong Vo, and Ba-Ngu Vo. A solution for large-scale multi-object tracking. *IEEE Transactions on Signal Processing*, 68:2754–2769, April 2020. doi:10.1109/TSP.2020.2986136.

[3] Benjamin L. Reifler and Brandon A. Jones. Tracking of large satellite constellations using a partitioned glmb filter. *Proceedings of the 2019 AAS/AIAA Astrodynamics Specialist Conference*, August 2019. Paper no. AAS 19-674.

[4] David W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley and Sons, New Jersey, NJ, 1992. doi:10.1002/0470045345.

[5] Jeffrey L. Anderson and Stephen L. Anderson. A monte carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Monthly Weather Review*, 127(12):2741–2758, 1999. doi:10.1175/1520-0493.

[6] Bo Liu, Boujemaa Ait-El-Fquih, and Ibrahim Hoteit. Efficient kernel-based ensemble gaussian mixture filtering. *Monthly Weather Review*, 144(2):781–800, 2016. doi:10.1175/MWR-D-14-00292.1.

[7] Sehyun Yun, Renato Zanetti, and Brandon A. Jones. Kernel-based ensemble gaussian mixture filtering for orbit determination with sparse data. *Proceedings of the 2021 AAS/AIAA Astrodynamics Specialist Conference*, 2021. Paper no. AAS 21-501.

[8] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002. doi:10.1109/78.978374.

[9] Bernard W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, London, 1986.

[10] Ba-Ngu Vo, Ba-Tuong Vo, and Hung Gia Hoang. An efficient implementation of the generalized labeled multi-Bernoulli filter. *IEEE Transactions on Signal Processing*, 65(8), April 2017. doi:10.1109/TSP.2016.2641392.

[11] Andreas Danzer, Stephan Reuter, and Klaus Dietmayer. The adaptive labeled multi-bernoulli filter. arXiv:1812.08790v1 [cs.SY], December 2018.