(Preprint) AAS 23-356

MCMC ENGMF FOR SPARSE DATA ORBIT DETERMINATION

Dalton Durant[*], Andrey A. Popov[†], and Renato Zanetti[‡]

There has been an increasing interest in orbit determination of the continued growing number of space objects to avoid collisions and monitor space activity. Accurate orbit determination relies on advanced filtering techniques and high quality measurements. Proposed in this work, MCMC EnGMF, uses a Markov Chain Monte Carlo approach to aggregate measurements from one track into a "measured" orbit solution whose uncertainty is expressed using a Gaussian Mixture Model. This enables the filter to take advantage of highly non-Gaussian orbit estimates arising from a single track. This work shows MCMC EnGMF is more robust under the challenging angles-only scenario and is more consistent than sequential filtering of raw measurements. Additionally, the relationships between the ensemble size of the processed measurement distribution and computational performance and accuracy are analyzed.

INTRODUCTION

The projected future increase in the number of space objects (SOs) is too large to track using current techniques and current ground stations. Either more hardware (ground stations and/or clusters of supercomputers) or increased computational efficiency of tracking software is needed. This work focuses on a software solution to improving orbit determination with sparse data, which inherits the problem of the current surveillance network's inability to follow all SOs at all times. Using the current number of ground stations in today's surveillance network, for a much larger SO population, results in fewer measurements for each SO. This subsequently results in tracking arcs that are shorter and/or sparser. In other words, measurements may only appear in dense sequences—tracks—that themselves are sparse due to visibility constraints of ground-based surveillance sensors.

Sparse and nonlinear measurements necessitate the development of nonlinear filters with non-Gaussian probability density function (PDF) approximations (*e.g.* EnGMF,[1–4] Particle filters,[5–7] EnKF,[8–11] AEGIS[12]). This paper modifies the Ensemble Gaussian Mixture Filter (EnGMF),[3] which outperformed the Unscented Kalman Filter (UKF)[13] and the Adaptive Entropy-based Gaussian-mixture Information Synthesis (AEGIS) filter[12] in terms of accuracy and consistency for orbit determination. It sequentially incorporates measurements from a track, and, at each measurement time, a prior PDF is formed using Silverman's rule of thumb.[14] This approach produces a non-divergent, but somewhat conservative solution. Therefore, tuning[14,15] or adaptation[4] can be performed to increase consistency and decrease conservativeness.

[*]Ph.D. Student, Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, TX 78712

[†]Postdoctoral Fellow, Oden Institute for Computational Engineering & Sciences, The University of Texas at Austin, Austin, TX 78712

[‡]Associate Professor, Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, TX 78712

This paper implements an alternative to tuning and adaptation for the EnGMF. Instead of sequentially filtering raw measurements in each track, this paper processes an entire track of raw measurements into one aggregate "measurement" in the state-space. Performed by a batch algorithm, this processed measurement results in a more Gaussian shaped measurement distribution opposed to the raw measurements filtered sequentially in time. This subsequently results in a more optimal Silverman's rule of thumb, which assumes Gaussian distributions for constructing the prior PDF of the EnGMF. This approach reduces the EnGMF conservativeness and retains its robustness properties.

But what happens when the measurement distribution cannot be assumed Gaussian? Tracking measurements are typically point-wise-in-time un-observable while a sequence of them is needed to reconstruct the orbit. A track consisting of observations of two angles, range, and range-rate provides good orbit reconstruction accuracy since the rate of the angles is strongly observable from the data. Hence, the aggregate measurement arising from such a track can be accurately represented with as Gaussian. A track consisting of angles-only measurements, on the other hand, retains significant range and range-rate uncertainty which, when coupled with their non-linearity, are not accurately represented by a Gaussian PDF but produce an unknown, non-Gaussian distribution. Therefore, a batch algorithm producing a single Gaussian distribution is not enough.

This paper presents the use of Markov Chain Monte Carlo (MCMC) methods to sample from unknown, non-Gaussian measurement distributions. A non-Gaussian distribution can be approximated by a Gaussian Mixture Model (GMM); a Kernel Density Estimation (KDE) method. So, with enough MCMC samples, an accurate approximation of the unknown, non-Gaussian distribution for the processed measurements can be built. The MCMC method used in this work is the Metropolis-Hastings (M-H) algorithm.[16] M-H is used to produce a GMM of processed measurements in the state-space from an unknown target distribution given a track of raw measurements.

The contributions of this paper can be summarized as follows:

- A joint MCMC and EnGMF algorithm (calling MCMC EnGMF) is proposed to efficiently track SOs in LEO with short and sparse observation data including the challenging angles-only scenario.
- The proposed MCMC EnGMF is compared to other state-of-the-art Monte Carlo approaches, EnGMF and Batch EnGMF, through numerical simulation in terms of accuracy, consistency, and computational speed.

This paper is organized as follows: First, the mechanics of the EnGMF, Batch EnGMF, and MCMC EnGMF will be presented along with some visual aids to promote understanding of their similarities and differences. Next, a results section which will go over the problem setup and review simulation results. These results include a comparison of filters for the case where raw measurements are observable across their measurement track and for the case when they are un-observable (angles-only). The results will also include an analysis of ensemble size for MCMC. Finally, a conclusion of the work and outlook of future work will be discussed.

ESTIMATION METHODS FOR NONLINEAR, NON-GAUSSIAN SYSTEMS

Consider a dynamic system represented by:

$$\begin{aligned} \mathrm{x}_k &= f(\mathrm{x}_{k-1}) + \eta_k, \\ \mathrm{y}_k &= h(\mathrm{x}_k) + \epsilon_k, \end{aligned} \tag{1}$$

where x_k is the state of the system and y_k is the measurement at time step k, and η_k and ϵ_k are Gaussian white noise processes with zero mean and covariance matrices Q and R, respectively. This paper assumes no process noise such that $\eta_k = \emptyset \; \forall \; k$.

The function $f(\mathrm{x}_{k-1})$ describes the nonlinear dynamics of the system and maps the previous state x_{k-1} to the current state x_k. Similarly, the function $h(\mathrm{x}_k)$ maps the current state x_k to the expected measurement y_k. This function may also be nonlinear.

The application of orbit determination complicates the state estimation problem into one that is nonlinear and non-Gaussian. Nonlinear in the sense that the system dynamics and measurements behave nonlinearly through time, and non-Gaussian in the sense that non-Gaussian distributions could arise during propagation between measurements, even though measurement errors can be assumed to have Gaussian distributions.

EnGMF

The Ensemble Gaussian Mixture Filter (EnGMF) is a Monte Carlo based method of filtering that utilizes a mixture of Gaussian distributions to model the state of a system where each Gaussian represents a hypothesis about the current state.[1,3] The Gaussian mixture is updated by applying a prediction step, which involves individually propagating each Gaussian forward in time according to the system dynamics, and an update step, which involves computing the likelihood of the measurements given each Gaussian and re-weighting the mixture accordingly. The EnGMF typically maintains a fixed number of Gaussians.

The EnGMF utilizes the concept of KDE;[14] a non-parametric method that estimates the probability density function of random variables.[17] The EnGMF uses KDE to combine two popular filtering techniques, *particle filters* and *Gaussian sum filters*; producing a smoothed representation of the distribution of particles, which in the limit of ensemble size converges to exact Bayesian inference. This can result in a more accurate representation of the posterior distribution, and can subsequently improve state estimation performance.[3]

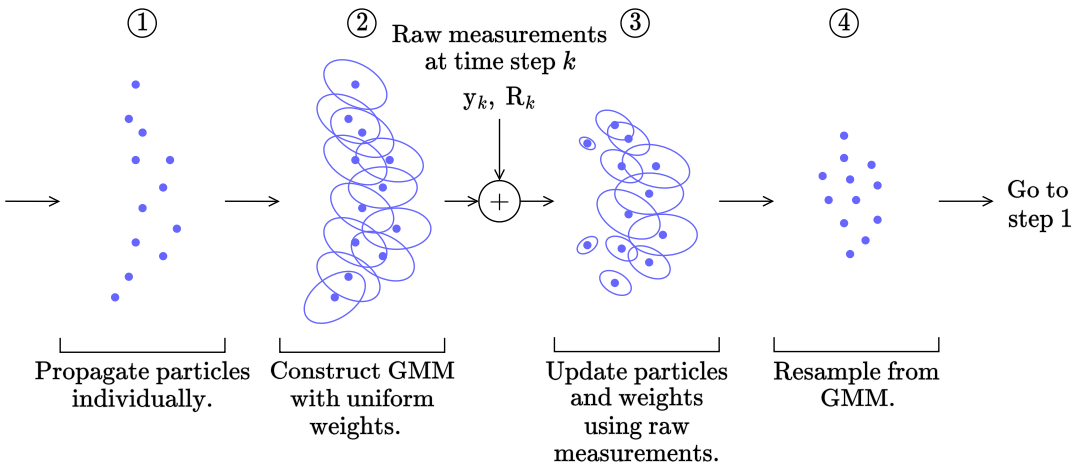

Figure 1: Visual aid of the EnGMF mechanics.

This work makes use of the Modified Kernel-Based Ensemble Gaussian Mixture Filtering method which is a sequential estimation filter that approximates the distribution of a system based on sparse

data.[3] The EnGMF assumes the knowledge of the distribution at the prior time, $\mathrm{p}(\mathrm{x}_{k-1})$, and approximates it with N independent and identically distributed (i.i.d.) samples (particles) $\mathrm{x}_{k-1}^{(i)}$,

$$\begin{aligned} \mathrm{p}(\mathrm{x}_{k-1}) &\approx \sum_{i=1}^{N} \frac{1}{N} \delta(\mathrm{x}_{k-1} - \mathrm{x}_{k-1}^{(i)}) \\ &\approx \lim_{\mathrm{P}_{k-1} \to O} \sum_{i=1}^{N} \frac{1}{N} \mathcal{N}(\mathrm{x}_{k-1}; \mathrm{x}_{k-1}^{(i)}, \mathrm{P}_{k-1}), \end{aligned} \tag{2}$$

where k is the discrete time step, $\delta(\cdot)$ is the Dirac delta distribution and can be said to be the normal distribution with covariance that tends towards the zero matrix O in the limit. The EnGMF only makes an empirical measure assumption just before the propagation step. This is represented by Equation (2) and occurs just before step 1 of Figure 1.

The particles are propagated to time step k, step 1 of Figure 1, and are then converted into Gaussian mixtures using a standard KDE method, where each particle is considered a Gaussian component with non-zero covariance of equal weights $\frac{1}{N}$. This is represented by Equation (3) and in step 2 of Figure 1. The approximated GMM of the propagated particles is:

$$\begin{aligned} \mathrm{p}(\mathrm{x}_k) &\approx \sum_{i=1}^{N} \frac{1}{N} \mathrm{p}(\mathrm{x}_k^{(i)}) \\ &\approx \sum_{i=1}^{N} \frac{1}{N} \mathcal{N}(\mathrm{x}_k; \bar{\mathrm{x}}_k^{(i)}, B), \end{aligned} \tag{3}$$

where $\bar{\mathrm{x}}_k^{(i)}$ is the propagated state of the i^{th} particle of time step k. The bandwidth matrix B is calculated by $B = \beta \bar{\mathrm{P}}_k$, where β is the bandwidth parameter, $\beta > 0$, and $\bar{\mathrm{P}}_k$ is the prior sample covariance.[2,3] The bandwidth parameter β is used to determine the covariance matrix of each Gaussian component. The larger the bandwidth parameter, the smaller the probability assigned to the particle and vice versa.* The GMM approximation of the posterior distribution is then updated using measurement information. This is represented in step 3 in Figure 1 and represented below:

$$\begin{aligned} \mathrm{p}(\mathrm{x}_k|\mathrm{y}_k) &\approx \sum_{i=1}^{N} w_k^{(i)} \mathrm{p}(\mathrm{y}_k|\mathrm{x}_k^{(i)}) \mathrm{p}(\mathrm{x}_k^{(i)}) \\ &\approx \sum_{i=1}^{N} w_k^{(i)} \mathcal{N}(\mathrm{x}_k; \hat{\mathrm{x}}_k^{(i)}, \hat{\mathrm{P}}_k^{(i)}). \end{aligned} \tag{4}$$

*The bandwidth parameter β is crucial in determining the performance of the filter, and its selection is a trade-off between accuracy and computational cost. This paper makes use of Silverman's rule of thumb to estimate the bandwidth matrix, which provides a near-optimal bandwidth parameter without the need for numerical optimization, assuming the sampling distribution is Gaussian.[3,14] However, when the distribution is not close to Gaussian, the estimates may be conservative (large). This is desirable as inaccuracies result in conservatism rather than over-confidence and divergence.

The approximate solution for analysis purposes is:

$$\hat{\mathrm{x}}_k \;=\; \sum_{i=1}^{N} w_k^{(i)} \hat{\mathrm{x}}_k^{(i)}, \tag{5}$$

$$\hat{\mathrm{P}}_k \;=\; \sum_{i=1}^{N} w_k^{(i)} [\hat{\mathrm{P}}_k^{(i)} + \hat{\mathrm{x}}_k^{(i)} (\hat{\mathrm{x}}_k^{(i)})^T - \hat{\mathrm{x}}\hat{\mathrm{x}}^T], \tag{6}$$

where $\hat{\mathrm{x}}_k^{(i)}$ and $\hat{\mathrm{P}}_k^{(i)}$ are the individual Extended Kalman Filter (EKF) state estimate and covariance estimate, respectively, for the i^{th} particle of time step k. The updated weights $w_k^{(i)}$ of each particle incorporate the raw measurements y_k and their covariance R_k. They are calculated below and sum to unity:[18]

$$w_k^{(i)} \;=\; \frac{w_{k-1}^{(i)} \mathcal{N}(\mathrm{y}_k; h_k(\bar{\mathrm{x}}_k^{(i)}), H_k^{(i)} B (H_k^{(i)})^T + \mathrm{R}_k)}{\sum_{i=1}^{N} w_{k-1}^{(i)} \mathcal{N}(\mathrm{y}_k; h_k(\bar{\mathrm{x}}_k^{(i)}), H_k^{(i)} B (H_k^{(i)})^T + \mathrm{R}_k)}, \tag{7}$$

$$w_k^{(i)} \;\geq\; 0, \;\; i \;=\; 1 \cdots N \qquad \sum_{i=1}^{N} w_k^{(i)} \;=\; 1, \tag{8}$$

where $H_k^{(i)}$ is the measurement Jacobian for the i^{th} particle.

The EnGMF then draws N i.i.d. samples from the GMM approximation of the posterior distribution.[18] This is represented in step 4 of Figure 1. These particles are propagated and are used as a starting point for the next iteration — Equation (2).

Batch EnGMF

The Batch EnGMF makes use of a batch processing algorithm that computes the Maximum *a Posteriori* (MAP) estimate of the state using all available raw measurements of a given pass. It condenses each track of raw measurements from the sensors into a single Gaussian distribution in the state-space. The distribution's mean and covariance are then fed into the EnGMF as "processed measurements".

This paper uses the batch least squares information filter.[19] It is based on the normal equations, which can be derived by setting the gradient of the cost function with respect to the state to zero and can be written as:

$$(H^T \mathrm{R}^{-1} H + \mathrm{P}_0^{-1}) \hat{\mathrm{x}} = H^T \mathrm{R}^{-1} \mathrm{y} + \mathrm{P}_0^{-1} \bar{\mathrm{x}}_0. \tag{9}$$

The solution to Equation (9) gives the MAP estimate of the state, $\hat{\mathrm{x}}$, where the state vector x is modeled as a Gaussian random variable with mean $\bar{\mathrm{x}}$ and covariance P. The measurement vector y is modeled as a noisy observation of the true state vector. The measurement model can be written as: $\mathrm{y} = H\mathrm{x} + v$ where H is the observation matrix and v is the measurement noise modeled as a Gaussian random variable with zero mean and covariance R. For a more detailed look at the batch least squares information filter see Tapley et al.[19]

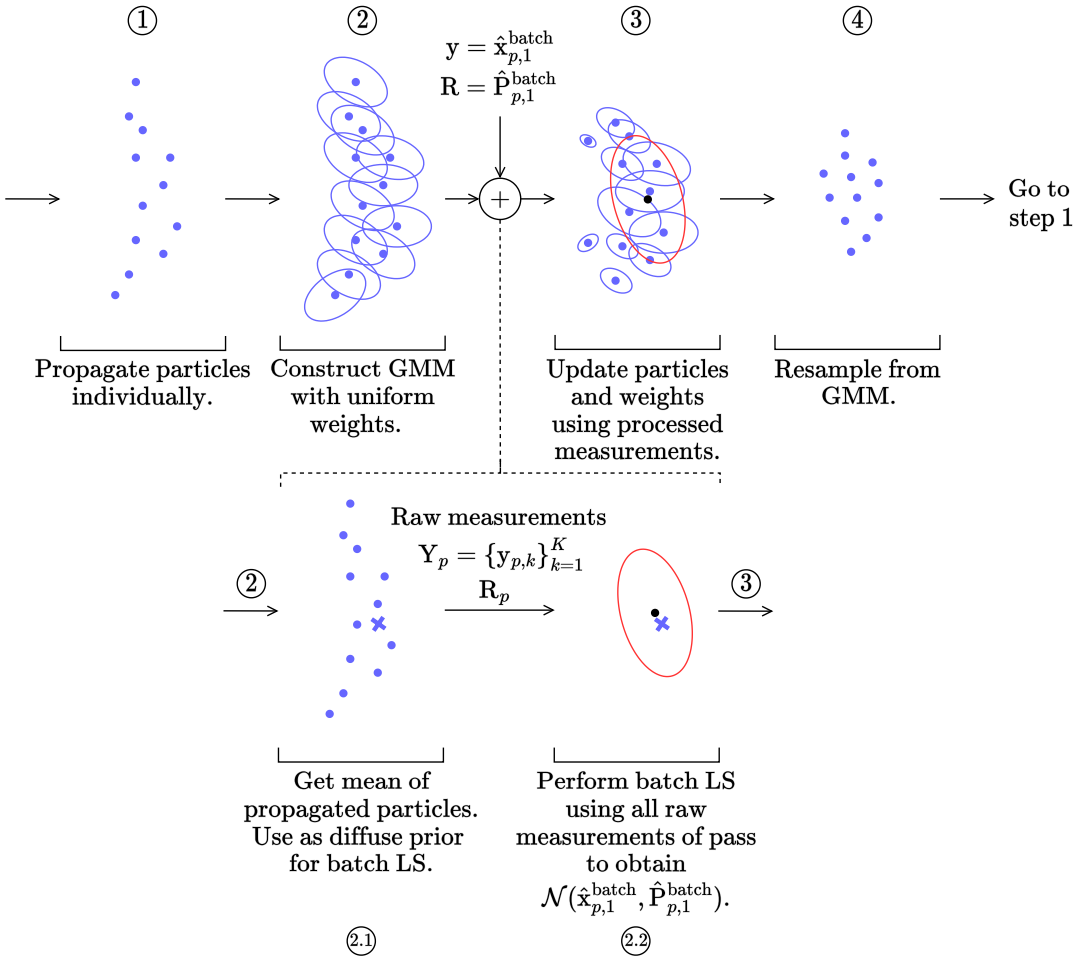

Figure 2: Visual aid of the Batch EnGMF mechanics.

In step 1 of Figure 2, the particles are individually propagated to the first time step of the next measurement pass p. The particles are then converted into Gaussian mixtures represented by Equation (10) and in step 2 of Figure 2. The approximated GMM of the propagated particles is a modification of Equation (3) such that:

$$p(x_p) \approx \sum_{i=1}^{N} \frac{1}{N} \mathcal{N}(x_p; \bar{x}_p^{(i)}, B), \tag{10}$$

where $\bar{x}_p^{(i)}$ is the propagated state of the i^{th} particle of pass p instead of time step k. So, instead of passing raw measurements sequentially, Batch EnGMF processes the entire track of a pass of raw measurements in aggregate. The raw measurements are stored as $Y_p = \{y_{p,k}\}_{k=1}^{K}$, R_p where k represents the k^{th} raw measurement of pass p. After a GMM is constructed, the sample mean of these particles is calculated and is fed into the batch least squares information filter as a diffuse prior* such that:

*A diffuse prior is a non-informative and flat prior that doesn't strongly restrict parameter values. It's useful when there's limited prior knowledge and lets the data strongly shape the posterior distribution.

$$\bar{\mathrm{x}}_{p,0}^{\text{batch}} \;=\; \sum_{i=1}^{N} w_p^{(i)} \bar{\mathrm{x}}_p^{(i)}, \tag{11}$$

$$(\bar{\mathrm{P}}_{p,0}^{\text{batch}})^{-1} \;=\; O, \tag{12}$$

where $\bar{\mathrm{x}}_{p,0}^{\text{batch}}$ is the prior sample mean with prior covariance $\bar{\mathrm{P}}_{p,0}^{\text{batch}}$. Additionally, $\bar{\mathrm{x}}_p^{(i)}$ is the propagated state for the i^{th} particle, and $w_p^{(i)} = \frac{1}{N}$ since all particles have uniform weight at this point. This is represented in step 2.1 of Figure 2.

Table 1: Batch Least Squares Information Filter

---

function batchLS_func($\bar{\mathrm{x}}_{p,0}^{\text{batch}}$, $\bar{\mathrm{P}}_{p,0}^{\text{batch}}$, Y_p, R_p, τ)
  INPUT: $\bar{\mathrm{x}}_{p,0}^{\text{batch}}$ initial guess, $\bar{\mathrm{P}}_{p,0}^{\text{batch}}$ initial covariance, Y_p set of raw measurements,
    R_p measurement covariance, τ convergence threshold.
  OUTPUT: $\mathrm{y} = \hat{\mathrm{x}}_1^{\text{batch}}$, $\mathrm{R} = \hat{\mathrm{P}}_1^{\text{batch}}$

  % Initialization
  $\delta\bar{\mathrm{x}}_1 = \emptyset$, $\hat{\mathrm{x}}_1 = \bar{\mathrm{x}}_{p,0}^{\text{batch}}$, rms $= 100$, rms$' = 0$
  while abs(rms $-$ rms$'$) $> \tau$ **do**
    $\bar{\mathrm{x}} = \hat{\mathrm{x}}_1$, $\Lambda_1 = (\bar{\mathrm{P}}_{p,0}^{\text{batch}})^{-1}$, $N_1 = \Lambda_1 \delta\bar{\mathrm{x}}_1$, $\Phi(1,1) = \mathbb{I}$ % Diffuse prior $(\bar{\mathrm{P}}_{p,0}^{\text{batch}})^{-1} = O$
    for $k = 1 : K$ **do**
      % Measurement update mapped to $k = 1$
      $\mathrm{y}_k = \mathrm{Y}_p^{(k)}$
      $\hat{\mathrm{y}}_k = h(\bar{\mathrm{x}})$
      $H_k = H(\bar{\mathrm{x}})$
      $\delta\mathrm{y}_k = \mathrm{y}_k - \hat{\mathrm{y}}_k$
      $\Lambda_1 = \Lambda_1 + \Phi(k,1)^T H_k^T \mathrm{R}_p^{-1} H_k \Phi(k,1)$
      $N_1 = N_1 + \Phi(k,1)^T H_k^T \mathrm{R}_p^{-1} \delta\mathrm{y}_k$
      if $k \neq K$ **then**
        $\bar{\mathrm{x}} = f(\bar{\mathrm{x}})_{k+1}$
        Compute state transition matrix $\Phi(k+1,k)$
        $\Phi(k+1,1) = \Phi(k+1,k)\Phi(k,1)$
      end if
    end for
    % Solve normal equations
    $\delta\hat{\mathrm{x}}_1 = \Lambda_1^{-1} N_1$
    $\hat{\mathrm{x}}_1 = \hat{\mathrm{x}}_1 + \delta\hat{\mathrm{x}}_1$
    $\delta\bar{\mathrm{x}}_1 = \delta\bar{\mathrm{x}}_1 - \delta\hat{\mathrm{x}}_1$
    rms $= \sqrt{\sum_{k=1}^{K} \frac{1}{|\mathrm{y}_k|} \delta\mathrm{y}_k^T \mathrm{R}_p^{-1} \delta\mathrm{y}_k}$
    rms$' =$ rms
  end while
  $\hat{\mathrm{x}}_1^{\text{batch}} = \hat{\mathrm{x}}_1$
  $\hat{\mathrm{P}}_1^{\text{batch}} = \Lambda_1^{-1}$
end *function*

Next, using all of the raw measurements of pass p mapped to the first raw measurement time

step $k = 1$, the batch filter produces a state estimate $\hat{\mathrm{x}}_{p,1}^{\text{batch}}$ and covariance estimate $\hat{\mathrm{P}}_{p,1}^{\text{batch}}$. This is represented in step 2.2 of Figure 2. Pseudocode of the batch least squares information filter is provided in Table 1.

The single batch estimate and covariance are then passed back into the EnGMF as processed measurements where each particle's individual EKF sees a measurement vector $\mathrm{y} = \hat{\mathrm{x}}_{p,1}^{\text{batch}}$, a measurement covariance $\mathrm{R} = \hat{\mathrm{P}}_{p,1}^{\text{batch}}$, and a measurement Jacobian $H = \mathbb{I}$; identity. The GMM approximation of the posterior distribution is then updated using the processed measurements. This is represented by Equation (13) and in step 3 of Figure 2:

$$\begin{aligned} \mathrm{p}(\mathrm{x}_p|\mathrm{y} = \hat{\mathrm{x}}_{p,1}^{\text{batch}}) &\approx \sum_{i=1}^{N} w_p^{(i)} \mathrm{p}(\mathrm{y} = \hat{\mathrm{x}}_{p,1}^{\text{batch}}|\mathrm{x}_p^{(i)}) \mathrm{p}(\mathrm{x}_p^{(i)}) \\ &\approx \sum_{i=1}^{N} w_p^{(i)} \mathcal{N}(\mathrm{x}_p; \hat{\mathrm{x}}_p^{(i)}, \hat{\mathrm{P}}_p^{(i)}). \end{aligned} \tag{13}$$

The approximate solution for analysis purposes is:

$$\hat{\mathrm{x}}_p = \sum_{i=1}^{N} w_p^{(i)} \hat{\mathrm{x}}_p^{(i)}, \tag{14}$$

$$\hat{\mathrm{P}}_p = \sum_{i=1}^{N} w_p^{(i)} [\hat{\mathrm{P}}_p^{(i)} + \hat{\mathrm{x}}_p^{(i)} (\hat{\mathrm{x}}_p^{(i)})^T - \hat{\mathrm{x}}\hat{\mathrm{x}}^T], \tag{15}$$

where $\hat{\mathrm{x}}_p^{(i)}$ and $\hat{\mathrm{P}}_p^{(i)}$ are the individual Extended Kalman Filter (EKF) state estimate and covariance estimate for the i^{th} particle using the batch estimate of pass p when $k = 1$. The updated weights $w_p^{(i)}$ of each particle incorporate the measurements $\mathrm{y} = \hat{\mathrm{x}}_{p,1}^{\text{batch}}$ and their covariance $\mathrm{R} = \hat{\mathrm{P}}_{p,1}^{\text{batch}}$. They are calculated using the following and sum to unity:

$$w_p^{(i)} = \frac{w_{p-1}^{(i)} \mathcal{N}(\mathrm{y}; h_p(\bar{\mathrm{x}}_p^{(i)}), B + \mathrm{R})}{\sum_{i=1}^{N} w_{p-1}^{(i)} \mathcal{N}(\mathrm{y}; h_p(\bar{\mathrm{x}}_p^{(i)}, B + \mathrm{R})}, \tag{16}$$

$$w_p^{(i)} \geq 0, \;\; i = 1 \cdots N \qquad \sum_{i=1}^{N} w_p^{(i)} = 1, \tag{17}$$

where $\bar{\mathrm{x}}_p^{(i)}$ is the dynamically propagated state of the i^{th} particle for pass p. The algorithm then draws N i.i.d. samples from the GMM approximation of the posterior distribution. These particles are used as a starting point for the next iteration. This is represented in step 4 of Figure 2.

MCMC EnGMF

The MCMC EnGMF is an EnGMF that uses a MCMC method to process raw measurements. This is similar to the Batch EnGMF except the batch least squares information filter is being replaced by

the M-H algorithm[16] to produce a GMM of processed measurements rather than a single Gaussian.* The M-H algorithm is tasked with producing an ensemble of new particles from an unknown target distribution given the raw measurements. This new ensemble is fed into the EnGMF as "processed measurements". By leveraging the strengths of MCMC and EnGMF, it is possible to achieve highly accurate and robust tracking of space objects in the presence of point-wise-in-time un-observable, noisy measurements.†

The Metropolis-Hastings (M-H) algorithm is a Markov Chain Monte Carlo (MCMC) method and is a powerful tool for exploring the probability space of complex models.[16,20] By proposing new states and accepting or rejecting them based on a probability distribution, the M-H algorithm can efficiently explore the probability space and generate samples from a target distribution which direct sampling is difficult. The algorithm is particularly useful for Bayesian inference, where the goal is to estimate the posterior distribution of a set of parameters given a set of measurements and a prior distribution.

Consider a set of measurements y and a set of parameters x that want to be estimated. Bayes' theorem states that the posterior distribution of x given y is equal to the product of the likelihood function of y given x, the prior distribution of x, and some constant c: $\mathrm{p(x|y)} = c\mathrm{p(y|x)p(x)}$. If the constant c is unknown, then the posterior distribution $\mathrm{p(x|y)}$ cannot be computed analytically. So, we must resort to MCMC methods such as the M-H algorithm to sample from $\mathrm{p(x|y)}$ because it only requires the ratio: $\mathrm{p(x^{'}|y)/p(x|y)} = c\mathrm{p(y|x^{'})p(x^{'})}/c\mathrm{p(y|x)p(x)} = \mathrm{p(y|x^{'})p(x^{'})/p(y|x)p(x)}$, where the constant c cancels out with itself. The M-H algorithm proceeds as follows:[16]

1. Choose an initial value for the parameters: $\mathrm{x} = \mathrm{x}_0$.

2. Propose a new value for the parameters $\mathrm{x}^{'}$ using a proposal distribution $\mathrm{q(x^{'}|x)}$.

3. Compute the acceptance probability $\alpha = \min\left\{1, \frac{\mathrm{p(x^{'}|y)\cdot q(x|x^{'})}}{\mathrm{p(x|y)\cdot q(x^{'}|x)}}\right\}$.

4. Generate a uniform random number u from the interval $[0, 1]$.

5. If $u \leq \alpha$, accept and store the proposal and set $\mathrm{x}_0 = \mathrm{x}^{'}$, otherwise reject the proposal and keep $\mathrm{x}_0 = \mathrm{x}$.

6. Repeat steps 2-5 until the desired number of samples is obtained.

*If a batch filter or any other linear filter was used to process the measurements, then it would only produce a Gaussian representation of the processed measurements and not a smooth one more akin to the actual non-Gaussian distribution.

†As seen later, the Batch EnGMF improves accuracy and consistency of the EnGMF for observable processed measurements. However, when the raw measurements are point-wise-in-time un-observable, the Batch EnGMF may also produce un-observable processed measurements. This results in inaccurate and conservative estimates.

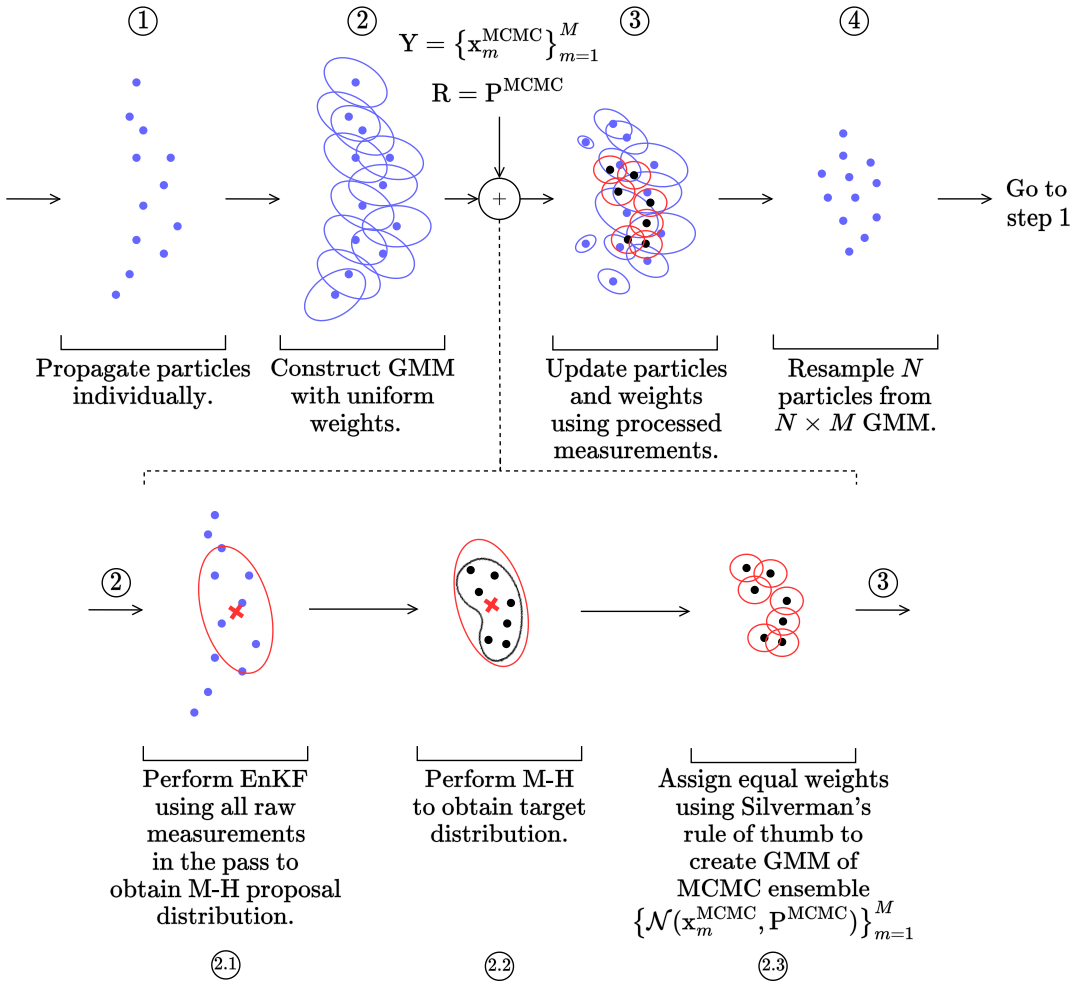

Figure 3: Visual aid of the MCMC EnGMF mechanics.

In step 1 of Figure 3, the particles are individually propagated to the first time step of the next measurement pass p. The particles are then converted into Gaussian mixtures represented by Equation (10) and in step 2 of Figure 3. After a GMM is constructed, M-H proposal distributions:

$$q(x^{'}|x) = \mathcal{N}(x^{'}; x, P_{p,k=1}^{EnKF}), \tag{18}$$

$$q(x|x^{'}) = \mathcal{N}(x; x^{'}, P_{p,k=1}^{EnKF}), \tag{19}$$

are formed using the Ensemble Kalman Filter (EnKF)[9,21,22] which produces a single Gaussian distribution where the initial guess for M-H $x_0 = \hat{x}_{p,k=1}^{EnKF}$ with covariance $P_{p,k=1}^{EnKF}$.* This is represented in step 2.1 of Figure 3. Next, for a given set of raw measurements, the M-H algorithm is performed to obtain the target distribution $p(x^{(m)}|Y_p)$ for each $m = 1 \cdots M$, where M is the desired size of the MCMC GMM ensemble. This is represented in step 2.2 of Figure 3. The target distribution

*Theoretically, this proposal distribution can be anything, however, the EnKF solution gives a good estimate of the target distribution and is efficient to compute.

used is:

$$\mathrm{p}(\mathrm{x}|\mathrm{Y}_p) \;\approx\; c\mathrm{p}(\mathrm{Y}_p|\mathrm{x}) \;=\; c\prod_{k=1}^{K}\exp\left(-[\mathrm{Y}_p^{(k)} - h(f(\mathrm{x})_k)]^T\mathrm{R}_p^{-1}[\mathrm{Y}_p^{(k)} - h(f(\mathrm{x})_k)]/2\right), \qquad (20)$$

where $\mathrm{p}(\mathrm{x}|\mathrm{Y}_p) \approx c\mathrm{p}(\mathrm{Y}_p|\mathrm{x})$ because the prior $\mathrm{p}(\mathrm{x})$ is assumed to be diffuse. As mentioned before, the unknown constant c in Equation (20) will cancel out with itself when computing the acceptance ratio α. Furthermore, Equation (20) represents the likelihood of the set of raw measurements Y_p given a proposed state x. In other words, it is the likelihood that the proposed Markov Chain state is a feasible realization (through time) for every raw measurement of the track. Can use Equation (20) to similarly compute $\mathrm{p}(\mathrm{x}^{'}|\mathrm{Y}_p)$ by swapping x with $\mathrm{x}^{'}$.

Table 2: MCMC Gaussian Mixture Model Algorithm

---

function mcmcgmm_func($\bar{\mathrm{X}}_p$, Y_p, R_p)
INPUT: $\bar{\mathrm{X}}_p$ propagated particles, Y_p set of raw measurements, R_p measurement covariance,
M size of MCMC GMM ensemble, L steps in Markov chain, maxiter max iterations.
OUTPUT: $\mathrm{Y} = \{\mathrm{x}_m^{\mathrm{MCMC}}\}_{m=1}^{M}, \mathrm{R} = \mathrm{P}^{\mathrm{MCMC}}$

% Perform Ensemble Kalman Filter
$[\hat{\mathrm{x}}_0,\ \hat{\mathrm{P}}_0] = \mathrm{enkf}(\bar{\mathrm{X}}_p,\ \mathrm{Y}_p,\ \mathrm{R}_p)$
for $m = 1 : M$ **do**
  acceptances $= 0$
  iter $= 0$
  % Perform Metropolis-Hastings for each m
  $\mathrm{x} = \hat{\mathrm{x}}_0$
  while acceptances $< L$ **&** iter $<$ maxiter **do**
    % Propose a new sample from the proposal distribution
    $\mathrm{x}^{'} \in \mathrm{q}(\mathrm{x}^{'}|\mathrm{x}) = \mathcal{N}(\mathrm{x}^{'}; \mathrm{x}, \hat{\mathrm{P}}_0)$
    % Compute distributions and acceptance ratio
    $\mathrm{q}(\mathrm{x}|\mathrm{x}^{'}) = \mathcal{N}(\mathrm{x}; \mathrm{x}^{'}, \hat{\mathrm{P}}_0)$
    $\mathrm{q}(\mathrm{x}^{'}|\mathrm{x}) = \mathcal{N}(\mathrm{x}^{'}; \mathrm{x}, \hat{\mathrm{P}}_0)$
    $\mathrm{p}(\mathrm{x}|\mathrm{Y}_p) = \prod_{k=1}^{K}\exp\left(-[\mathrm{Y}_p^{(k)} - h(f(\mathrm{x})_k)]^T\mathrm{R}_p^{-1}[\mathrm{Y}_p^{(k)} - h(f(\mathrm{x})_k)]/2\right)$
    $\mathrm{p}(\mathrm{x}^{'}|\mathrm{Y}_p) = \prod_{k=1}^{K}\exp\left(-[\mathrm{Y}_p^{(k)} - h(f(\mathrm{x}^{'})_k)]^T\mathrm{R}_p^{-1}[\mathrm{Y}_p^{(k)} - h(f(\mathrm{x}^{'})_k)]/2\right)$
    $\alpha = \min\left(1,\ \frac{\mathrm{p}(\mathrm{x}^{'}|\mathrm{Y}_p)}{\mathrm{p}(\mathrm{x}|\mathrm{Y}_p)}\frac{\mathrm{q}(\mathrm{x}|\mathrm{x}^{'})}{\mathrm{q}(\mathrm{x}^{'}|\mathrm{x})}\right)$
    if $(u \in [0, 1]) < \alpha$ **then**
      $\mathrm{x} = \mathrm{x}^{'}$
      acceptances $+= 1$
    end if
    iter $+= 1$
  end while
  % Save last state of Markov Chain into GMM ensemble
  $\mathrm{x}_m^{\mathrm{MCMC}} = \mathrm{x}$
end for
Compute sample covariance of ensemble $\hat{\mathrm{P}}^{\mathrm{MCMC}}$
Compute bandwidth parameter β^{MCMC}
$\mathrm{P}^{\mathrm{MCMC}} = B^{\mathrm{MCMC}} = \beta^{\mathrm{MCMC}}\hat{\mathrm{P}}^{\mathrm{MCMC}}$
end *function*

The MCMC particles are then converted into Gaussian mixtures represented by Equation (21) and in step 2.3 of Figure 3:

$$\mathrm{p}(\mathrm{x}^{\mathrm{MCMC}}) \approx \sum_{m=1}^{M} \frac{1}{M} \mathcal{N}(\mathrm{x}^{\mathrm{MCMC}}; \mathrm{x}_m^{\mathrm{MCMC}}, B^{\mathrm{MCMC}}). \tag{21}$$

The bandwidth matrix B^{MCMC} is calculated by $B^{\mathrm{MCMC}} = \beta^{\mathrm{MCMC}} \hat{\mathrm{P}}^{\mathrm{MCMC}}$, where $\hat{\mathrm{P}}^{\mathrm{MCMC}}$ is the sample covariance and β^{MCMC} is the bandwidth parameter and is calculated using Silverman's rule of thumb, but instead for M particles.

Instead of producing a single Gaussian state and covariance estimate seen with the Batch En-GMF, the M-H algorithm produces a GMM that is fed into the EnGMF as processed measurements. Pseudocode of the algorithm used to perform M-H and generate a GMM of size M is provided in Table 2.*† This enables state estimation for systems that are both nonlinear and non-Gaussian, especially when measurements are point-wise-in-time un-observable. For N particles, $i = 1, \cdots, N$, each particle's individual EKF sees M separate measurement vectors $\mathrm{y}^{(m)} = \mathrm{x}_m^{\mathrm{MCMC}}$, each with a measurement covariance $\mathrm{R} = B^{\mathrm{MCMC}}$, and a measurement Jacobian $H = \mathbb{I}$; identity. This will construct a new GMM that is size $N \times M$. The GMM approximation of the posterior distribution is then updated using measurement information. This is represented by Equation (22) and in step 3 of Figure 3:

$$\begin{aligned} \mathrm{p}(\mathrm{x}_p|\mathrm{Y}) &\approx \sum_{i=1}^{N} \sum_{m=1}^{M} w_p^{(i,m)} \mathrm{p}(\mathrm{x}_{\mathrm{m}}^{\mathrm{MCMC}}|\mathrm{x}_p^{(i)}) \mathrm{p}(\mathrm{x}_p^{(i)}) \\ &\approx \sum_{i=1}^{N} \sum_{m=1}^{M} w_p^{(i,m)} \mathcal{N}(\mathrm{x}_p; \hat{\mathrm{x}}_p^{(i,m)}, \hat{\mathrm{P}}_p^{(i,m)}). \end{aligned} \tag{22}$$

The approximate solution for analysis purposes is:

$$\hat{\mathrm{x}}_p = \sum_{i=1}^{N} \sum_{m=1}^{M} w_p^{(i,m)} \hat{\mathrm{x}}_p^{(i,m)}, \tag{23}$$

$$\hat{\mathrm{P}}_p = \sum_{i=1}^{N} \sum_{m=1}^{M} w_p^{(i,m)} [\hat{\mathrm{P}}_p^{(i,m)} + \hat{\mathrm{x}}_p^{(i,m)} (\hat{\mathrm{x}}_p^{(i,m)})^T - \hat{\mathrm{x}}\hat{\mathrm{x}}^T], \tag{24}$$

where $\hat{\mathrm{x}}_p^{(i,m)}$ and $\hat{\mathrm{P}}_p^{(i,m)}$ are the individual Extended Kalman Filter (EKF) state and covariance estimate, respectively, for the i^{th} particle using the m^{th} MCMC processed measurement of pass p. The updated weights $w_p^{(i,m)}$ of the particles incorporate the measurements $\mathrm{Y} = \{\mathrm{x}_m^{\mathrm{MCMC}} : m =$

*When the algorithm performs M-H for each m, the **while** loop continues until the user defined maximum Markov Chain steps L or until the maximum loop iterations maxiter; whichever comes first. This is done for computational efficiency since performing all L steps can be expensive and sometimes unnecessary to obtain a good realization of the posterior distribution, so maxiter is used to terminate the loop at the current Markov Chain step. This paper used $L = 10$ and maxiter $= 100$.

†Notice that $\mathrm{q}(\mathrm{x}|\mathrm{x}') = \mathrm{q}(\mathrm{x}'|\mathrm{x})$. Any proposal that satisfies this is called "symmetric" and therefore the Metropolis-Hastings algorithm boils down to just the Metropolis algorithm[20] where the acceptance ratio is just $\alpha = \min(1, \frac{\mathrm{p}(\mathrm{x}'|\mathrm{y})}{\mathrm{p}(\mathrm{x}|\mathrm{y})})$.

$1 \cdots M\}$ and their covariance $\mathrm{R} = B^{\mathrm{MCMC}}$. They are calculated using the following and sum to unity:

$$w_p^{(i,m)} = \frac{w_{p-1}^{(i,m)} \mathcal{N}(\mathrm{y}^{(m)}; h_p(\bar{\mathrm{x}}_p^{(i)}), B + \mathrm{R})}{\sum_{i=1}^{N} \sum_{m=1}^{M} w_{p-1}^{(i,m)} \mathcal{N}(\mathrm{y}^{(m)}; h_p(\bar{\mathrm{x}}_p^{(i)}, B + \mathrm{R})}, \tag{25}$$

$$w_p^{(i,m)} \geq 0, \ i = 1 \cdots N, \ m = 1 \cdots M \qquad \sum_{i=1}^{N} \sum_{m=1}^{M} w_p^{(i,m)} = 1, \tag{26}$$

where $\bar{\mathrm{x}}_p^{(i)}$ is the dynamically propagated state of the i^{th} particle for pass p. The algorithm then draws N i.i.d. samples from the $N \times M$ GMM approximation of the posterior distribution using the method described in (Reference 18). These particles are used as a starting point for the next iteration. This is represented in step 4 in Figure 3.

Table 3: General Ensemble Gaussian Mixture Filter

function engmf_func($\bar{\mathrm{X}}$, Y, R)
  INPUT: $\bar{\mathrm{X}}$ set of N propagated particles, Y measurements either raw or processed,
    R measurement covariance.
  OUTPUT: $\bar{\mathrm{X}} = \{\bar{\mathrm{x}}^{(i)}\}_{i=1}^{N}$

  Compute bandwidth matrix B
  % Set weights to be uniform
  $w^{(i,m)} = 1/(N \times M) \ \ \forall\, i = 1 \cdots N, \ m = 1 \cdots M$
  % Perform independent Extended Kalman Filter update for each particle
  $\hat{\mathrm{x}}^{(i,m)} = \emptyset, \ \hat{\mathrm{P}}^{(i,m)} = \emptyset \ \ \forall\, i = 1 \cdots N, \ m = 1 \cdots M$
  for $i = 1 : N$ **do**
    for $m = 1 : M$ **do** % $M = 1$ for EnGMF and Batch EnGMF
      $P_{xx} = B$
      $P_{xy} = P_{xx} H(\bar{\mathrm{x}}^{(i)})^T$ % $H(\cdot) = \mathbb{I}$ for Batch EnGMF and MCMC EnGMF
      $P_{yy} = H(\bar{\mathrm{x}}^{(i)}) P_{xx} H(\bar{\mathrm{x}}^{(i)})^T + \mathrm{R}$
      $K = P_{xy} P_{yy}^{-1}$
      $\hat{\mathrm{x}}^{(i,m)} = \bar{\mathrm{x}}^{(i)} + K(\mathrm{Y}^{(m)} - h(\bar{\mathrm{x}}^{(i)}))$ % $h(\mathrm{x}) = \mathrm{x}$ for Batch EnGMF and MCMC EnGMF
      $\hat{\mathrm{P}}^{(i,m)} = (\mathbb{I} - KH(\bar{\mathrm{x}}^{(i)})) P_{xx} (\mathbb{I} - KH(\bar{\mathrm{x}}^{(i)}))^T + K\mathrm{R}K^T$
      $w^{(i,m)} = w^{(i,m)} \mathcal{N}(\mathrm{Y}^{(m)}; h(\bar{\mathrm{x}}^{(i)}), P_{yy})$
    end for
  end for
  Enforce Equation (26) requirements for the $N \times M$ weights
  Sample N particles from posterior GMM using method described in Yun et al.[18] $\{\mathrm{x}^{(i)}\}_{i=1}^{N}$
  % Propagate resampled particles individually through dynamics
  for $i = 1 : N$ **do**
    $\bar{\mathrm{x}}^{(i)} = f(\mathrm{x}^{(i)})$
  end for
end *function*

Pseudocode of the generalized EnGMF is provided in Table 3 which can accommodate raw and processed measurements. If using the EnGMF or Batch EnGMF, then the ensemble size $M = 1$

since there is no GMM of processed measurements. Additionally, if using Batch EnGMF or MCMC EnGMF, then the measurement function $h(\mathrm{x}) = \mathrm{x}$ and it's jacobian $H(\cdot) = \mathbb{I}$ since the processed measurements exist in the state-space.

RESULTS

Problem Setup

Parameters. To evaluate the performance of the EnGMF, Batch EnGMF, and MCMC EnGMF, one numerical example is considered. The parameters are taken from Yun et al.[3] to perform a direct comparison to their modified EnGMF. The system dynamic equations are numerically integrated with an embedded Runge-Kutta 8(7) method.[23] Range, range-rate, and angle measurements are simulated using a ground station located at the North Pole (latitude = 90°, longitude = 0°, altitude = 0 km). Using a fictitious tracking station at the north pole allows for a LEO SO in near-polar orbit to be visible every orbit. Hence a sensitivity study of the accuracy of the tracking algorithms as a function of the frequency of the measurement passes (once per orbit, once every two orbits, etc.) is easily performed without the need of scheduling sensors at different geographic locations to insure the desired measurement frequency. In this simulation, tracking passes are short and sparse. The measurements are available every 10 seconds with a pass lasting only 2 minutes (*i.e.* 12 measurements per pass).

The SO is in a near polar orbit with the following Keplerian orbital elements:

$$\begin{aligned} a &= 7,078.0068 \text{ [km]} \\ e &= 0.01 \\ i &= 85^\circ \\ \Omega &= 0^\circ \\ \omega &= 0^\circ \\ \nu &= 0^\circ, \end{aligned} \tag{27}$$

where a is the semi-major axis, e is the eccentricity, i is the inclination, Ω is the longitude of the ascending node, ω is the argument of periapsis, and ν is the true anomaly. Additionally, the mass of the SO is 500 kg.

The state vector is size 6×1 where $\mathrm{x} = [r, v]^T$ and r is a 3×1 position vector and v is a 3×1 velocity vector. The initial distribution is defined in Cartesian coordinates as:

$$\mathrm{x}_0 \sim \mathcal{N}(\bar{\mathrm{x}}_0, \mathrm{P}_0), \tag{28}$$

$$\bar{\mathrm{x}}_0 = \begin{bmatrix} 7078.0068 & \text{[km]} \\ 0 & \text{[km]} \\ 0 & \text{[km]} \\ 0 & \text{[km/s]} \\ 0.6606 & \text{[km/s]} \\ 7.5509 & \text{[km/s]} \end{bmatrix}, \tag{29}$$

$$\mathrm{P}_0 = \begin{bmatrix} 1.481\mathrm{e}2 & 0 & 0 & 0 & -9.237\mathrm{e}{-2} & -5.333\mathrm{e}{-2} \\ 0 & 2.885\mathrm{e}1 & 9.994 & -3.232\mathrm{e}{-2} & 0 & 0 \\ 0 & 9.994 & 5.770 & -1.242\mathrm{e}{-2} & 0 & 0 \\ 0 & -3.232\mathrm{e}{-2} & -1.242\mathrm{e}{-2} & 3.687\mathrm{e}{-5} & 0 & 0 \\ -9.237\mathrm{e}{-2} & 0 & 0 & 0 & 6.798\mathrm{e}{-5} & 3.145\mathrm{e}{-5} \\ -5.333\mathrm{e}{-2} & 0 & 0 & 0 & 3.145\mathrm{e}{-5} & 3.166\mathrm{e}{-5} \end{bmatrix}. \quad (30)$$

This initial distribution begins at the equator and is propagated over the north pole for the system to observe it's first track of raw measurements. The starting time of each measurement pass is randomly selected in close proximity of a multiple of the orbital period.

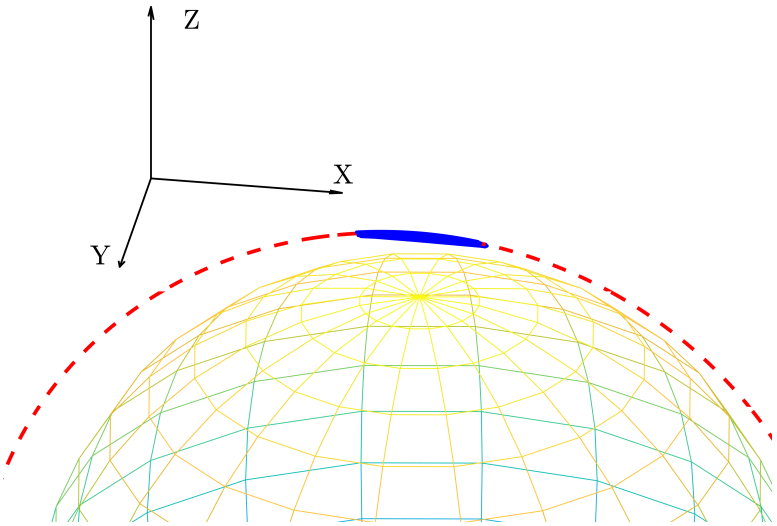

Figure 4: The orbit of the SO in the ECI frame.

The orbit of the SO is shown in Figure 4. Each filter is ran with 1000 particles for 100 Monte Carlo iterations with a constant number of orbital periods between measurement tracks for 60 orbits ($\sim$ 100 hours). For example, if the orbital period between measurement tracks was 5, then measurements would be filtered every 5 orbits. Each filter was ran with the same truth data and changed only the number of orbital periods between measurement tracks for different test cases. This was performed for skipping between 1–10 orbits. By changing the number of orbital periods between measurement passes, we can showcase each filter's ability to estimate sparse measurements. Additionally, unless otherwise stated, the MCMC EnGMF's M-H algorithm uses 200 particles in it's target distribution ($M = 200$). This was chosen as a heuristic and is explained later.

Dynamics model. The state vector $\mathrm{x} = [\boldsymbol{r^I}, \boldsymbol{v^I}]^T$ is constructed by $\boldsymbol{r^I} = [r_x, r_y, r_z]^T$ and $\boldsymbol{v^I} = [v_x, v_y, v_z]^T$ which are the Earth-Centered Inertial (ECI) position and velocity coordinates, respectively. The orbital dynamics are:

$$\begin{bmatrix} \dot{\boldsymbol{r}}^{\boldsymbol{I}} \\ \dot{\boldsymbol{v}}^{\boldsymbol{I}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{v^I} \\ -\frac{\mu}{\|r\|_2^3}\boldsymbol{r^I} \end{bmatrix}, \quad (31)$$

where μ is the Earth's gravitational parameter and $\| \ r \ \|_2$ is the Euclidean norm of $\boldsymbol{r^I}$. This work only uses 2-body dynamics.

Measurement model. The raw measurement vector $\mathrm{y} = [\rho,\ \dot{\rho},\ \alpha,\ \delta]^T$ contains range ρ, range-rate $\dot{\rho}$, right ascension α, and declination δ of the observed SO as seen by a ground-based radar sensor. The range and range-rate measurements are calculated along the line of sight (LOS) between the ground-based radar station and the SO:

$$\rho \ = \| \ \boldsymbol{r^I} \ - \ \boldsymbol{r^I_S} \ \|_2, \tag{32}$$

$$\dot{\rho} \ = \ \frac{(\boldsymbol{r^I} \ - \ \boldsymbol{r^I_S})^T(\boldsymbol{v^I} \ - \ \boldsymbol{v^I_S})}{\rho}, \tag{33}$$

where $\boldsymbol{r^I_S} = [r_{S_x}, r_{S_y}, r_{S_z}]^T$ and $\boldsymbol{v^I_S} = [v_{S_x}, v_{S_y}, v_{S_z}]^T$ are the position and velocity, respectively, of the ground station in the inertial frame. Additionally, right ascension α and declination δ are:

$$\alpha \ = \ \tan^{-1}\left(\frac{r_y \ - \ r_{S_y}}{r_x \ - \ r_{S_x}}\right), \tag{34}$$

$$\delta \ = \ \sin^{-1}\left(\frac{r_z \ - \ r_{S_z}}{\rho}\right). \tag{35}$$

In this study, light travel time delay and measurement biases are not considered. The measurements are corrupted by additive zero-mean Gaussian white noise with standard deviation of 30 m and 0.3 m/s for the range and range-rate, respectively, and 100 arc-seconds on the right ascension and declination observations.

Simulation Results

The EnGMF, Batch EnGMF, and MCMC EnGMF are compared based on accuracy, consistency, and computational performance. The accuracy of the filters is examined by their root-mean-square error (RMSE), which is computed from the true and estimated states at each measurement update time for all Monte Carlo simulations. The RMSE is defined as follows:

$$\mathrm{RMSE}_k \ = \ \frac{1}{\mathcal{M}} \sum_{i=1}^{\mathcal{M}} \sqrt{\frac{1}{n_\mathrm{x}} (\mathrm{x}_k^{(i)} - \hat{\mathrm{x}}_k^{(i)})^T (\mathrm{x}_k^{(i)} - \hat{\mathrm{x}}_k^{(i)})}. \tag{36}$$

Additionally, the consistency of the filters is examined using the scaled normalized estimation error squared (SNEES) which is defined as follows:[24]

$$\mathrm{SNEES}_k \ = \ \frac{1}{\mathcal{M} n_\mathrm{x}} \sum_{i=1}^{\mathcal{M}} (\mathrm{x}_k^{(i)} - \hat{\mathrm{x}}_k^{(i)})^T (\hat{\mathrm{P}}_k^{(i)})^{-1} (\mathrm{x}_k^{(i)} - \hat{\mathrm{x}}_k^{(i)}), \tag{37}$$

where $\mathcal{M}$ is the number of Monte Carlo iterations, n_x is the size of the state-space, $\mathrm{x}_k^{(i)}$ are the true states, $\hat{\mathrm{x}}_k^{(i)}$ are the estimated states, and $\hat{\mathrm{P}}_k^{(i)}$ is the estimated covariance of the i^{th} Monte Carlo iteration at time step/pass k (depending on which filter being used). An ideal RMSE is one that converges to zero. Anything greater than zero suggests inaccuracies with respect to the truth. The size of the state-space $n_\mathrm{x} = 6$ is used to scale the NEES value.[24] An ideal SNEES is equal to 1. Anything less than 1 tells that a filter is conservative and anything greater than 1 tells that a filter is too confident. Computational performance is examined by computing each filter's average execution time to complete 100 Monte Carlo iterations. This project used Matlab with an 8-core CPU and ran each Monte Carlo iteration in parallel to utilize all 8 cores.

Full measurement vector. The following results utilize a full raw measurement vector:

$$\mathrm{y} = [\rho,\ \dot{\rho},\ \alpha,\ \delta]^T. \tag{38}$$

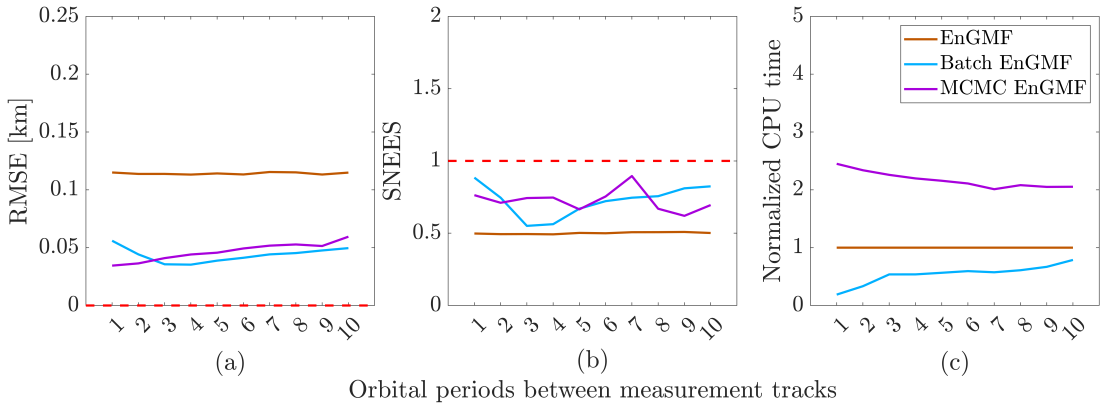

Figure 5: For a full measurement vector — Equation (38) — Monte Carlo averaged position RMSE (a), SNEES (b), & CPU time normalized with respect to EnGMF CPU time (c).

Figure 5 displays the Monte Carlo averaged results for each filter using the full measurement vector. Figure 5 (a) shows the RMSE of each filter averaged over all time (60 orbits) for all Monte Carlo iterations (100) skipping orbital periods between tracks for each case. Both the Batch EnGMF and MCMC EnGMF have similar performance with the Batch EnGMF performing slightly better. They both outperform the modified EnGMF and reduce the RMSE by half.

Figure 5 (b) shows the filter consistency by computing the SNEES of each filter averaged over all time (60 orbits) for all Monte Carlo iterations (100) for each orbit skipping case. Both the Batch EnGMF and MCMC EnGMF prove to be more consistent than the EnGMF which is fairly conservative.

Figure 5 (c) shows the normalized CPU time of each filter averaged over all time (60 orbits) for all Monte Carlo iterations (100) for each orbit skipping case. Each filter's runtime is normalized to the EnGMF's runtime for comparison purposes. The Batch EnGMF proves to be about 2 times quicker whereas the MCMC EnGMF is 2 times slower. The EnGMF accuracy, consistency, and computational performance results here are reminiscent of the results from Yun et al.[3]

Varying the MCMC GMM ensemble size M. The following results also utilize a full measurement vector, Equation (38). However, the ensemble size of the MCMC GMM produced by the M-H algorithm for MCMC EnGMF is varied across each case. This was performed for 100 Monte Carlo iterations with 1000 particles in the main ensemble (N=1000). Additionally, each case is skipping 10 orbits between tracks. The only parameter being varied across each case is the MCMC ensemble size (M). This analysis is performed to understand why the heuristic $M = 200$ was chosen for the previous full measurement vector results and for the later angles-only results.

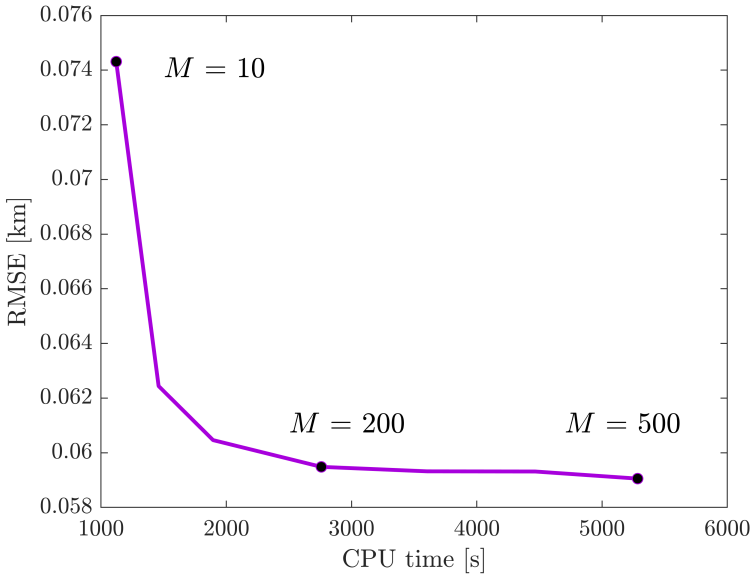

Figure 6: For a full measurement vector — Equation (38) — Monte Carlo and time averaged RMSE as a function of CPU time for varying MCMC ensemble sizes M.

Figure 6 displays the time averaged Monte Carlo results for varying M in the MCMC EnGMF using the full measurement vector. The number of particles does not aid much in filter accuracy after the MCMC ensemble size grows larger than ~ 100 ($M \geq 100$). To exist within a safe balance of accuracy and performance, this research uses $M = 200$ as a good heuristic.

Angles-only measurements. The full measurement vector case proved that the Batch EnGMF was better in accuracy, consistency, and computational performance than the EnGMF. However, the system was observable across sequential raw measurements in each track. This means the processed measurement distributions were okay to be assumed as single Gaussians. However, this can become problematic for an un-observable system where this distribution is actually highly non-Gaussian. The MCMC EnGMF performs well for un-observable systems (*e.g.* angles-only raw measurements) because of its ability to treat the measurement update target distribution as a GMM rather than a single Gaussian; encapsulating the nonlinearities of the distribution. The following results utilize angles-only raw measurement vectors:

$$\mathrm{y} = [\alpha,\ \delta]^T. \tag{39}$$

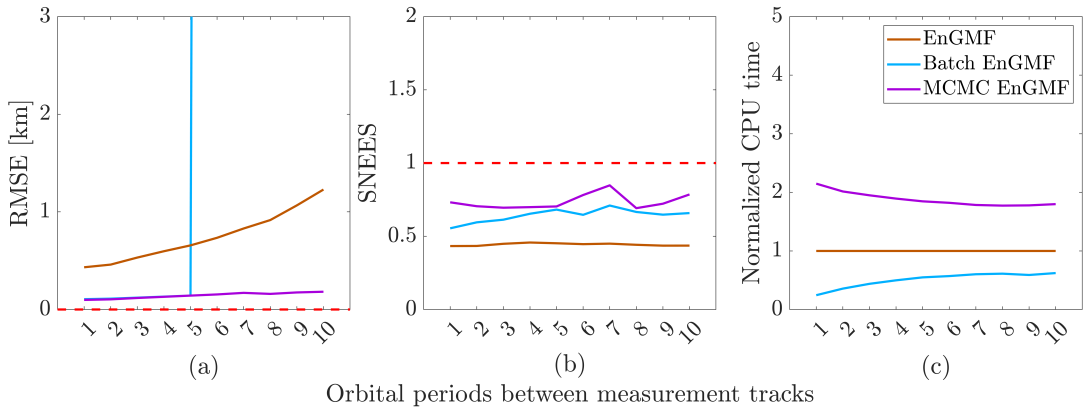

Figure 7: For a angles-only measurement vector — Equation (39) — Monte Carlo averaged position RMSE (a), SNEES (b), & CPU time normalized with respect to EnGMF CPU time (c).

Figure 7 displays the time averaged Monte Carlo results for each filter using the angles-only measurement vector. Figure 7 (a) shows the RMSE of each filter averaged over all time (60 orbits) for all Monte Carlo iterations (100) for skipping orbital periods between tracks for each case. Both the Batch EnGMF and MCMC EnGMF have similar performance until the number of orbital periods between tracks becomes greater than 4. The Batch EnGMF time averaged RMSE diverges rapidly, which suggests that approximating the processed measurements with a single Gaussian is not enough as time between orbits increases. The Batch EnGMF fails. Even though the EnGMF does not diverge rapidly, it does decrease in accuracy (plot increasing slope) faster than the MCMC EnGMF. Figure 7 (a) shows that the MCMC EnGMF performs much better in accuracy than both EnGMF and Batch EnGMF.

Figure 7 (b) shows the filter consistency by computing the SNEES of each filter averaged over all time (60 orbits) for all Monte Carlo iterations (100) for each orbit skipping case. Both the Batch EnGMF and MCMC EnGMF prove to be more consistent than the EnGMF which is still fairly conservative. However, the Batch EnGMF consistency in this plot is meritless since it's accuracy diverges rapidly in Figure 7 (a). Figure 7 (b) shows that the MCMC EnGMF performs much better in consistency than both EnGMF and Batch EnGMF.

Figure 7 (c) shows the normalized CPU time of each filter averaged over all time (60 orbits) for all Monte Carlo iterations (100) for each orbit skipping case. Each filter's runtime is normalized to the EnGMF's runtime for comparison purposes. The Batch EnGMF proves to be about 2 times quicker, but this performance is again less meaningful since it's accuracy diverges rapidly in Figure 7 (a). The MCMC EnGMF is $\sim 2\times$ slower than the EnGMF, but this suggests there is a risk-reward associated with the MCMC EnGMF: *performance is sacrificed for accuracy and consistency*. These results were also obtained using $M = 200$ for the MCMC ensemble size. What was shown previously is that decreasing M can improve computational performance with the caveat that decreasing too much can hurt accuracy.

CONCLUSION

This paper compares sequential Monte Carlo approaches to filtering for SO orbit determination. The filters in this paper use a standard KDE method, which involves using a sum of Gaussian kernels to create a continuous representation of the prior probability distribution from random samples. This allows us to represent non-Gaussian probability distributions and therefore fully utilize the advantages of a Monte Carlo approach. Sequential raw measurements of orbit tracks are processed into a single Gaussian distribution for Batch EnGMF and processed into a sum of Gaussians through a GMM for MCMC EnGMF. It is shown that observability issues with the Batch EnGMF arise for the angles-only case when approximating the processed measurement distribution as a single Gaussian. The estimation filter proposed in this paper, MCMC EnGMF, circumvents these observability issues by approximating this distribution as a GMM using the MCMC based M-H algorithm. Accuracy and consistency of the MCMC EnGMF prove to be considerably better than both the EnGMF^3 and the Batch EnGMF. Additionally, an analysis of varying the ensemble size of the outputted M-H's GMM is performed; expressing a relationship between ensemble size and computational performance, and ensemble size and accuracy. Future work in this research area can be to consider effects of perturbations (*e.g.*, J2, SRP, drag) and to improve computational speed of the MCMC EnGMF with possible other methods of MCMC other than the M-H algorithm.

ACKNOWLEDGMENT

This material is based on research sponsored by Air Force Research Laboratory (AFRL) under agreement number FA9550-22-1-0419, *Predictive Digital Twins at Scale for Space Systems*. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements of the US Government.

REFERENCES

[1] J. L. Anderson and S. L. Anderson, "A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts," *Monthly weather review*, Vol. 127, No. 12, 1999, pp. 2741–2758.

[2] B. Liu, B. Ait-El-Fquih, and I. Hoteit, "Efficient kernel-based ensemble Gaussian mixture filtering," *Monthly Weather Review*, Vol. 144, No. 2, 2016, pp. 781–800.

[3] S. Yun, R. Zanetti, and B. A. Jones, "Kernel-based ensemble gaussian mixture filtering for orbit determination with sparse data," *Advances in Space Research*, Vol. 69, No. 12, 2022, pp. 4179–4197.

[4] A. A. Popov and R. Zanetti, "An Adaptive Covariance Parameterization Technique for the Ensemble Gaussian Mixture Filter," *arXiv preprint arXiv:2212.10323*, 2022.

[5] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications Partical Filters for Tracking Applications.*, Vol. 830. Artec House, 2004.

[6] A. Doucet, N. De Freitas, N. J. Gordon, *et al.*, *Sequential Monte Carlo methods in practice*, Vol. 1. Springer, 2001.

[7] A. Doucet, A. M. Johansen, *et al.*, "A tutorial on particle filtering and smoothing: Fifteen years later," *Handbook of nonlinear filtering*, Vol. 12, No. 656-704, 2009, p. 3.

[8] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic engineering*, Vol. 82, No. 1, 1960, pp. 35–45.

[9] G. Evensen, "The ensemble Kalman filter: theoretical formulation and practical implementation," *Ocean dynamics*, Vol. 53, No. 4, 2003, pp. 343–367.

[10] M. Roth, G. Hendeby, C. Fritsche, and F. Gustafsson, "The Ensemble Kalman filter: a signal processing perspective," *EURASIP Journal on Advances in Signal Processing*, Vol. 2017, 2017, pp. 1–16.

[11] A. Hommels, A. Murakami, and S.-I. Nishimura, "A comparison of the ensemble Kalman filter with the unscented Kalman filter: application to the construction of a road embankment," *Geotechniek*, Vol. 13, No. 1, 2009, p. 52.

[12] K. J. DeMars, R. H. Bishop, and M. K. Jah, "Entropy-based approach for uncertainty propagation of nonlinear dynamical systems," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 4, 2013, pp. 1047–1057.

[13] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, Vol. 92, No. 3, 2004, pp. 401–422.

[14] B. W. Silverman, *Density estimation for statistics and data analysis*, Vol. 26. CRC press, 1986.

[15] P. Janssen, J. S. Marron, N. Veraverbeke, and W. Sarle, "Scale measures for bandwidth selection," *Journaltitle of Nonparametric Statistics*, Vol. 5, No. 4, 1995, pp. 359–380.

[16] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, 1970.

[17] D. W. Scott, *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[18] S. Yun and R. Zanetti, "Sequential monte carlo filtering with gaussian mixture sampling," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 9, 2019, pp. 2069–2077.

[19] B. T. B. Schutz, G. Born, *Statistical Orbit Determination*. Elsevier Academic Press, 2004.

[20] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The journal of chemical physics*, Vol. 21, No. 6, 1953, pp. 1087–1092.

[21] G. Evensen, "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics," *Journal of Geophysical Research: Oceans*, Vol. 99, No. C5, 1994, pp. 10143–10162.

[22] G. Burgers, P. J. Van Leeuwen, and G. Evensen, "Analysis scheme in the ensemble Kalman filter," *Monthly weather review*, Vol. 126, No. 6, 1998, pp. 1719–1724.

[23] J. R. Dormand and P. J. Prince, "A family of embedded Runge-Kutta formulae," *Journal of computational and applied mathematics*, Vol. 6, No. 1, 1980, pp. 19–26.

[24] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2001.