Gaussian Mixture-Based Point Mass Filtering with Applications to Terrain-Relative Navigation

FELIPE GIRALDO-GRUESO
The University of Texas at Austin, Austin, TX

ANDREY A. POPOV
The University of Texas at Austin, Austin, TX

RENATO ZANETTI, Senior Member, IEEE
The University of Texas at Austin, Austin, TX

***Abstract*— The accuracy of the point mass filter (PMF) relies on the precise placement of grid points. Since the approximated probability distributions are evaluated only at these points, suboptimal grid placement can result in an inaccurate representation of the posterior distribution. This work addresses this issue by introducing a variant of the PMF that represents the propagated grid points as a Gaussian mixture, enabling a Gaussian sum filter (GSF) update before grid construction. The GSF update improves the accuracy of the posterior mean and covariance estimates, leading to better grid placement. Additionally, an extension is presented, using kernel density estimation techniques to improve filter performance in low process noise scenarios. A comparative analysis is conducted between the proposed approach, the standard PMF, and other PMF variants. Using a bivariate example, the proposed method shows a better approximation of the posterior distribution compared to the other filters. Furthermore, two sequential filtering problems are used to analyze the performance of the filter, the first involving the Ikeda map and the second focusing on terrain-relative navigation. The results show that the proposed method provides more accurate and consistent filtering compared to the other PMF variants considered.**

***Index Terms*— Nonlinear Estimation, Point Mass Filters, Gaussian Mixtures, Grid Design**

Manuscript received XXXXX 00, 0000; revised XXXXX 00, 0000; accepted XXXXX 00, 0000.
This material is based on research sponsored by Air Force Office of Scientific Research (AFOSR) under award number: FA9550-22-1-0419.

Felipe Giraldo-Grueso is with the Department of Aerospace Engineering & Engineering Mechanics at The University of Texas at Austin, Austin, TX 78705 USA (e-mail: fgiraldo@utexas.edu). Andrey A. Popov was with the Oden Institute for Computational Sciences & Engineering at The University of Texas at Austin, Austin, TX 78705 USA (e-mail: andrey.a.popov@utexas.edu). Renato Zanetti is with the the Department of Aerospace Engineering & Engineering Mechanics and the Oden Institute at The University of Texas at Austin, Austin, TX 78705 USA (e-mail: renato@utexas.edu).

I. INTRODUCTION

Estimating quantities of interest from dynamic models and noisy measurements poses a significant challenge, especially in the context of nonlinear systems. A key approach to addressing this challenge involves solving the Bayesian recursive relations (BRRs) [1]. In these relations, the initial state probability density function (pdf) is propagated using the Chapman-Kolmogorov equation (in the context of discrete-time stochastic dynamic systems). Once a measurement is obtained, the pdf is updated using Bayes' rule. However, these relations can become analytically intractable when dealing with highly nonlinear dynamics and measurement models [1]. Consequently, approximations to the solution of the BRRs become necessary.

The point mass filter (PMF) represents a common methodology for numerically solving the BRRs [2]. This type of filter approximates the BRRs solution through deterministic grid-based integration techniques. The PMF approximates pdfs only at the grid points, making the placement of these points essential for its performance. Therefore, most of the PMF research has concentrated on refining grid design. Different approaches have focused on optimizing the predictive grid using an anticipative grid design technique [3], [4]. Although these strategies have proven to improve the performance of the PMF, incorporating the measurement to construct the predictive grid has the potential to yield a more accurate approximation of the posterior pdf.

Recent works have studied the idea of incorporating measurements as additional information when creating a predictive grid. Bergman [5], [6] proposed to contract the grid by removing points with low probability after performing the measurement update. Duník et al. [7] introduced an algorithm that uses an auxiliary unscented Kalman filter (UKF) to estimate the first two moments of the prior and posterior distributions in order to increase grid resolution in the approximated non-negligible support of these pdfs. Similarly, Choe and Park [8] developed an algorithm that uses log-homotopy induced flow to derive a grid support that accurately represents the non-negligible region of the posterior distribution.

In this work, a new approach is introduced for generating the predictive grid, called the fusion mass filter (FMF).[1] In the FMF, the propagated points from the initial filtering grid are represented as a Gaussian mixture, enabling a Gaussian sum filter (GSF) update prior to grid construction. This leads to a more accurate estimation of the mean and covariance of the posterior pdf, similar to the techniques used in the ensemble Gaussian mixture filter (EnGMF) [10]–[13]. This information is then used to construct an improved predictive grid. The application of this new technique results in tighter grids, which

[1]This paper is an extension to the work presented at the 27th International Conference on Information Fusion [9].

require fewer grid points to achieve accurate estimation performance.

In addition to presenting the FMF, this work has been expanded to include the following new contributions. First, two different weighting schemes used to obtain the filtering grid from the predictive grid in the FMF are compared. Second, an extension to the FMF, referred to as the Silverman mass filter (SMF), is introduced. In the SMF, the predictive distribution is approximated using kernel density estimation techniques, which improves performance in lower process noise scenarios. Lastly, a new terrain-relative navigation (TRN) algorithm is presented, where the position of a vehicle is estimated nonlinearly using the presented filters, while the velocity is estimated linearly using the extended Kalman filter (EKF) equations.

To demonstrate the performance of this new approach, a bivariate example is used first, showing that the new technique can approximate the posterior distribution more accurately than other PMF variants. The new filter is also validated using two sequential filtering scenarios. The first example uses the Ikeda map to evaluate the performance of the filter in a chaotic system. The second example focuses on TRN for Martian exploration. The TRN estimation framework integrates the new filter with the Q-method EKF update [14] to estimate both position and velocity simultaneously. Using this update technique, the states used in the measurement model can be updated with a nonlinear filter, such as PMFs, while the remaining states can be updated linearly, as in a standard EKF.

The remainder of this paper is organized as follows: first, a background of state estimation in discrete systems is provided in Section II. The standard PMF is introduced in Section III, the FMF is presented in Section IV, and the SMF is presented in Section V. A graphic comparison between the FMF, SMF and different variants of the PMF is illustrated in Section VI using a bivariate example. The performance of the filters is tested on a sequential filtering problem using the Ikeda map in Section VII. A closer look at the SMF is presented in Section VIII. The filters are evaluated in a more practical example with respect to TRN in Section IX. Finally, Section X provides conclusions and future work.

II. STATE ESTIMATION FOR DISCRETE SYSTEMS

In this section, background on state estimation for discrete systems is provided. Given the evolution of a state $\boldsymbol{x}_k$ governed by dynamics $\boldsymbol{f}_k$ and measurements $\boldsymbol{h}_k$,

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_k\left(\boldsymbol{x}_k, \boldsymbol{q}_k\right), \tag{1}$$

$$\boldsymbol{y}_k = \boldsymbol{h}_k\left(\boldsymbol{x}_k, \boldsymbol{\eta}_k\right), \tag{2}$$

where $\boldsymbol{y}_k$ is the associated measurement, $\boldsymbol{q}_k$ is the process noise and $\boldsymbol{\eta}_k$ is the measurement noise, state estimation deals with determining the time evolution of the pdf of the state given the dynamic knowledge and measurements obtained.

The solution to the state estimation problem is obtained by solving the BRRs. Starting from an initial pdf, the Chapman-Kolmogorov equation is used to propagate this pdf over time [1],

$$p\left(\boldsymbol{x}_{k+1}|\boldsymbol{y}_k\right) = \int_{\mathcal{S}(\boldsymbol{x}_k)} p\left(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k\right) p\left(\boldsymbol{x}_k|\boldsymbol{y}_k\right)\, d\boldsymbol{x}_k, \tag{3}$$

where $\mathcal{S}\left(\boldsymbol{x}_k\right)$ denotes the support of $\boldsymbol{x}_k$. Once a measurement is obtained ($k \leftarrow k+1$), the pdf is updated using Bayes' rule [1],

$$p\left(\boldsymbol{x}_k|\boldsymbol{y}_k\right) = \frac{p\left(\boldsymbol{y}_k|\boldsymbol{x}_k, \boldsymbol{y}_{k-1}\right) p\left(\boldsymbol{x}_k|\boldsymbol{y}_{k-1}\right)}{p\left(\boldsymbol{y}_k|\boldsymbol{y}_{k-1}\right)}. \tag{4}$$

The solution to these relations can become intractable when dealing with highly nonlinear dynamics or measurements. In such cases, state estimation involves implementing efficient and accurate approximations to iteratively solve these two equations until a desired time is reached.

III. POINT MASS FILTER

The PMF aims to solve the state estimation problem deterministically through a structured grid of point particles. Despite the increased computational overhead, this filter has shown superior estimation accuracy compared to standard particle filters, especially in cases involving distributions with heavy tails [4]. For this filtering strategy, instead of describing the state pdf with realizations of exchangeable samples (as done in particle filters), the state pdf is discretized by strategically placing a finite number of possible realizations at deterministic grid points, where each point is assigned a finite probability.

In the PMF, the initial posterior pdf can be approximated as a Dirac mixture such that [2]

$$p\left(\boldsymbol{x}_k|\boldsymbol{y}_k\right) \approx \sum_{i=1}^{N} w_{k|k}^{(i)} \delta\left(\boldsymbol{x}_k - \boldsymbol{\mathcal{X}}_{k|k}^{(i)}\right), \tag{5}$$

where $\boldsymbol{\mathcal{X}}_{k|k}^{(i)}$ are the posterior discretization points, $w_{k|k}^{(i)}$ represents the probability of each point, and N is the total number of points. Using the Chapman-Kolmogorov equation, the approximated pdf is then propagated,

$$p\left(\boldsymbol{x}_{k+1}|\boldsymbol{y}_k\right) \approx \int_{\mathcal{S}(\boldsymbol{x}_k)} p\left(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k\right) \sum_{i=1}^{N} w_{k|k}^{(i)} \delta\left(\boldsymbol{x}_k - \boldsymbol{\mathcal{X}}_{k|k}^{(i)}\right) d\boldsymbol{x}_k, \tag{6}$$

$$= \sum_{i=1}^{N} w_{k|k}^{(i)} \int_{\mathcal{S}(\boldsymbol{x}_k)} p\left(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k\right) \delta\left(\boldsymbol{x}_k - \boldsymbol{\mathcal{X}}_{k|k}^{(i)}\right)\, d\boldsymbol{x}_k, \tag{7}$$

$$= \sum_{i=1}^{N} w_{k|k}^{(i)}\, p\left(\boldsymbol{x}_{k+1}|\boldsymbol{\mathcal{X}}_{k|k}^{(i)}\right). \tag{8}$$

Using a Dirac mixture approximation, the resulting predictive distribution can be expressed as

$$p\left(\boldsymbol{x}_{k+1}|\boldsymbol{y}_k\right) \approx \sum_{j=1}^{M} w_{k+1|k}^{(j)} \delta\left(\boldsymbol{x}_{k+1} - \boldsymbol{\mathcal{X}}_{k+1}^{(j)}\right), \tag{9}$$

where $\boldsymbol{\mathcal{X}}_{k+1}^{(j)}$ are the M new discretization points, and the propagated weights are proportional to

$$w_{k+1|k}^{(j)} \propto \sum_{i=1}^{N} w_{k|k}^{(i)}\, p\left(\boldsymbol{\mathcal{X}}_{k+1}^{(j)} | \boldsymbol{\mathcal{X}}_{k|k}^{(i)}\right). \tag{10}$$

It is important to note that the number of grid points, M, in the discretization of the support of $\boldsymbol{x}_{k+1}$, does not have to be the same as in the initial discretization, N, of the support of $\boldsymbol{x}_k$. To generate the discretization points for the predictive pdf, a new uniform grid is constructed, referred to as the predictive grid. The center of the grid is obtained by calculating the mean of the propagated points, while the orientation and expanse of the grid are given by the propagated covariance. In the case of zero-mean additive process noise with covariance Q_k, the mean and covariance of the propagated points are given by [2]

$$\hat{\boldsymbol{x}}_{k+1|k} = \sum_{i=1}^{N} w_{k|k}^{(i)} \boldsymbol{f}_k\left(\boldsymbol{\mathcal{X}}_{k|k}^{(i)}\right), \tag{11}$$

$$P_{k+1|k} = Q_k - \hat{\boldsymbol{x}}_{k+1|k}\hat{\boldsymbol{x}}_{k+1|k}^{\mathrm{T}} + \sum_{i=1}^{N} w_{k|k}^{(i)} \left[\boldsymbol{f}_k\left(\boldsymbol{\mathcal{X}}_{k|k}^{(i)}\right) \boldsymbol{f}_k\left(\boldsymbol{\mathcal{X}}_{k|k}^{(i)}\right)^{\mathrm{T}}\right]. \tag{12}$$

Once the new grid has been generated and a new measurement has been obtained ($k \leftarrow k+1$), the weights of the grid are updated to generate the filtering grid. Using Bayes' rule,

$$p\left(\boldsymbol{x}_k|\boldsymbol{y}_k\right) \propto p\left(\boldsymbol{y}_k|\boldsymbol{x}_k\right) p\left(\boldsymbol{x}_k|\boldsymbol{y}_{k-1}\right), \tag{13}$$

$$\approx \sum_{j=1}^{M} w_{k|k-1}^{(j)} p\left(\boldsymbol{y}_k|\boldsymbol{x}_k\right) \delta\left(\boldsymbol{x}_k - \boldsymbol{\mathcal{X}}_k^{(j)}\right), \tag{14}$$

$$= \sum_{j=1}^{M} w_{k|k}^{(j)} \delta\left(\boldsymbol{x}_k - \boldsymbol{\mathcal{X}}_k^{(j)}\right), \tag{15}$$

where the new weights are proportional to the likelihood of measurement,

$$w_{k|k}^{(j)} \propto w_{k|k-1}^{(j)} p\left(\boldsymbol{y}_k|\boldsymbol{\mathcal{X}}_k^{(j)}\right). \tag{16}$$

The new weighted points then become the posterior discretization points, such that

$$\boldsymbol{\mathcal{X}}_{k|k}^{(j)} = \boldsymbol{\mathcal{X}}_k^{(j)}, \tag{17}$$

where the first two moments of the approximated posterior pdf are calculated as

$$\hat{\boldsymbol{x}}_{k|k} = \sum_{j=1}^{M} w_{k|k}^{(j)} \boldsymbol{\mathcal{X}}_{k|k}^{(j)}, \tag{18}$$

$$P_{k|k} = \sum_{j=1}^{M} w_{k|k}^{(j)} \left[\boldsymbol{\mathcal{X}}_{k|k}^{(j)} \left(\boldsymbol{\mathcal{X}}_{k|k}^{(j)}\right)^{\mathrm{T}}\right] - \hat{\boldsymbol{x}}_{k|k}\hat{\boldsymbol{x}}_{k|k}^{\mathrm{T}}. \tag{19}$$

These equations summarize the standard PMF, which achieves accurate and consistent results when the grid separation is less than the standard deviation of the process noise [15]. In such cases, the convolution step enables the weight to spread from points with high probability to their neighboring points. However, in scenarios characterized by low process noise, the filter may lead to grid collapse, concentrating all probability in a single point, which can result in denser grids and increased computational costs. In Fig. 1, a comparison between high and low process noise scenarios can be seen.

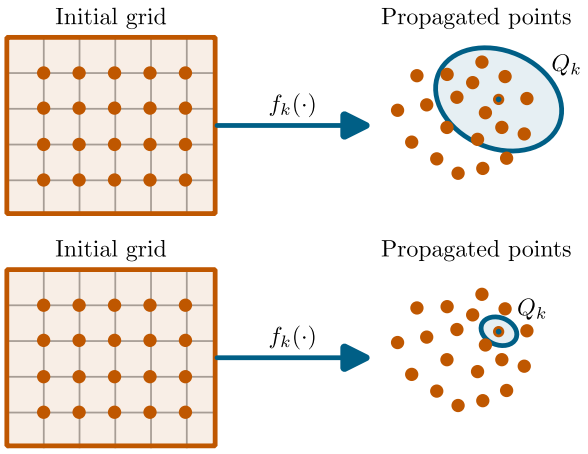

Fig. 1. Comparison between high (top) and low (bottom) process noise scenarios.

IV. FUSION MASS FILTER

This section introduces a new method for creating the predictive grid in the PMF, referred to as the FMF since it leverages information from both dynamics and measurements to construct a more accurate grid. This approach begins with a similar approximation to the PMF, by discretizing the initial posterior state pdf as a Dirac mixture. However, it introduces a subtle distinction by reformulating the Dirac mixture as a Gaussian mixture with infinitesimal covariance,

$$p\left(\boldsymbol{x}_k|\boldsymbol{y}_k\right) \approx \sum_{i=1}^{N} w_{k|k}^{(i)} \delta\left(\boldsymbol{x}_k - \boldsymbol{\mathcal{X}}_{k|k}^{(i)}\right), \tag{20}$$

$$= \sum_{i=1}^{N} w_{k|k}^{(i)} \lim_{P\to 0} \mathcal{N}\left(\boldsymbol{x}_k; \boldsymbol{\mathcal{X}}_{k|k}^{(i)}, P\right). \tag{21}$$

Assuming zero-mean additive process noise with covariance Q_k and using the GSF algorithm [16], [17], the resulting Gaussian mixture can be propagated to approximate the predictive distribution as

$$p\left(\boldsymbol{x}_{k+1}|\boldsymbol{y}_k\right) \approx \sum_{i=1}^{N} w_{k+1|k}^{(i)} \mathcal{N}\left(\boldsymbol{x}_{k+1}; \boldsymbol{f}_k\left(\boldsymbol{\mathcal{X}}_{k|k}^{(i)}\right), Q_k\right), \tag{22}$$

where

$$w_{k+1|k}^{(i)} = w_{k|k}^{(i)}. \tag{23}$$

Instead of constructing a predictive grid directly from the propagated points, this new approach uses a GSF update prior to assembling the predictive grid. This process has the potential to improve the estimation of the mean and covariance of the state pdf, thus improving the placement of the grid.

Once a measurement arrives ($k \leftarrow k+1$), the posterior distribution can be approximated from the prior Gaussian mixture [16], [17] as

$$p\left(\boldsymbol{x}_k|\boldsymbol{y}_k\right) \approx \sum_{i=1}^{N} \widetilde{w}_{k|k}^{(i)} \mathcal{N}\left(\boldsymbol{x}_k; \widetilde{\boldsymbol{\mathcal{X}}}_{k|k}^{(i)}, \widetilde{\mathcal{P}}_{k|k}^{(i)}\right), \tag{24}$$

where the notation $\widetilde{\cdot}$ means that these are auxiliary variables obtained with the GSF update. Each mean and covariance of the Gaussian mixture are given by

$$\widetilde{\boldsymbol{\mathcal{X}}}_{k|k}^{(i)} = \boldsymbol{\mathcal{X}}_{k|k-1}^{(i)} + K_k^{(i)} \boldsymbol{\nu}_k^{(i)}, \tag{25}$$

$$\widetilde{\mathcal{P}}_{k|k}^{(i)} = Q_{k-1} - K_k^{(i)} W_k^{(i)} K_k^{(i)\mathrm{T}}, \tag{26}$$

with the intermediate variables defined as

$$\boldsymbol{\mathcal{X}}_{k|k-1}^{(i)} = \boldsymbol{f}_{k-1}\left(\boldsymbol{\mathcal{X}}_{k-1|k-1}^{(i)}\right), \tag{27}$$

$$\boldsymbol{\nu}_k^{(i)} = \boldsymbol{y}_k - \boldsymbol{h}_k\left(\boldsymbol{\mathcal{X}}_{k|k-1}^{(i)}\right), \tag{28}$$

$$K_k^{(i)} = Q_{k-1} H_k^{(i)\mathrm{T}} \left(W_k^{(i)}\right)^{-1}, \tag{29}$$

$$W_k^{(i)} = H_k^{(i)} Q_{k-1} H_k^{(i)\mathrm{T}} + R_k, \tag{30}$$

$$H_k^{(i)} = \left.\frac{\partial \boldsymbol{h}_k\left(\boldsymbol{x}\right)}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{\mathcal{X}}_{k|k-1}^{(i)}}. \tag{31}$$

The weights of this new Gaussian mixture are defined as proportional to the probability of the obtained measurement,

$$\widetilde{w}_{k|k}^{(i)} \propto w_{k|k-1}^{(i)} \mathcal{N}\left(\boldsymbol{y}_k; \boldsymbol{h}_k\left(\boldsymbol{\mathcal{X}}_{k|k-1}^{(i)}\right), W_k^{(i)}\right). \tag{32}$$

Rather than creating a predictive grid with the propagated mean and covariance estimates, this approach discretizes the support based on the approximation of the posterior pdf. The predictive grid is assembled after the GSF update, using both the updated mean and covariance estimates, given by

$$\widetilde{\boldsymbol{x}}_{k|k} = \sum_{i=1}^{N} \widetilde{w}_{k|k}^{(i)} \widetilde{\boldsymbol{\mathcal{X}}}_{k|k}^{(i)}, \tag{33}$$

$$\widetilde{P}_{k|k} = \sum_{i=1}^{N} \widetilde{w}_{k|k}^{(i)} \left(\widetilde{\mathcal{P}}_{k|k}^{(i)} + \widetilde{\boldsymbol{\mathcal{X}}}_{k|k}^{(i)} \left(\widetilde{\boldsymbol{\mathcal{X}}}_{k|k}^{(i)}\right)^{\mathrm{T}} - \widetilde{\boldsymbol{x}}_{k|k} \widetilde{\boldsymbol{x}}_{k|k}^{\mathrm{T}}\right), \tag{34}$$

where the center of the grid is placed at the updated mean and the orientation and expanse are set to align with the updated covariance. With the new grid, the posterior distribution is re-approximated as a Dirac mixture,

$$p\left(\boldsymbol{x}_k|\boldsymbol{y}_k\right) \approx \sum_{j=1}^{M} w_{k|k}^{(j)} \delta\left(\boldsymbol{x}_k - \boldsymbol{\mathcal{X}}_k^{(j)}\right), \tag{35}$$

where $\boldsymbol{\mathcal{X}}_k^{(j)}$ are the new grid points, and $w_{k|k}^{(j)}$ are the new weights for each point.

To calculate these new weights and generate the filtering grid, two different strategies can be used. For the first one, the posterior pdf is expressed as

$$p\left(\boldsymbol{x}_k|\boldsymbol{y}_k\right) = \int_{\mathcal{S}(\boldsymbol{x}_{k-1})} p\left(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}|\boldsymbol{y}_k\right) \, d\boldsymbol{x}_{k-1}, \tag{36}$$

where

$$p\left(\boldsymbol{x}_k, \boldsymbol{x}_{k-1}|\boldsymbol{y}_k\right) \propto p\left(\boldsymbol{y}_k|\boldsymbol{x}_k\right) p\left(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}\right) \cdot \ldots \cdot p\left(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{k-1}\right). \tag{37}$$

Which results in weights proportional to

$$w_{k|k}^{(j)} \propto p\left(\boldsymbol{\mathcal{X}}_k^{(j)}|\boldsymbol{y}_k\right), \tag{38}$$

$$\approx \sum_{i=1}^{N} p\left(\boldsymbol{\mathcal{X}}_k^{(j)}, \boldsymbol{\mathcal{X}}_{k-1|k-1}^{(i)}|\boldsymbol{y}_k\right), \tag{39}$$

$$= p\left(\boldsymbol{y}_k|\boldsymbol{\mathcal{X}}_k^{(j)}\right) \sum_{i=1}^{N} p\left(\boldsymbol{\mathcal{X}}_k^{(j)}|\boldsymbol{\mathcal{X}}_{k-1|k-1}^{(i)}\right) w_{k-1|k-1}^{(i)}. \tag{40}$$

The filter using this weighting scheme is referred to as FMF1. The second strategy takes advantage of the weights from the GSF update such that

$$w_{k|k}^{(j)} \propto \sum_{i=1}^{N} \tilde{w}_{k|k}^{(i)} \mathcal{N}\left(\boldsymbol{\mathcal{X}}_k^{(j)}; \tilde{\boldsymbol{\mathcal{X}}}_{k|k}^{(i)}, \tilde{\mathcal{P}}_{k|k}^{(i)}\right). \tag{41}$$

The filter using this second weighting scheme is referred to as FMF2. Note that (41) converges to the particle filter as $N \to \infty$, thus converging to the true posterior. Just as in the PMF, the new weighted points become the posterior discretization points, and the first two moments of the approximated posterior pdf are calculated as in (18) and (19).

These equations summarize the new methodology for constructing a more precise predictive grid within the PMF framework, potentially improving the accuracy and consistency of the filter. Figure 2 presents a comparative analysis between the generation of the predictive grid in the PMF and in the proposed FMF.

V. SILVERMAN MASS FILTER

Different strategies have been proposed to address the small process noise problem in the PMF, including density-weighted convolution [15] and moment-matched Gaussian kernels [18]. In this section, an extension to the FMF is presented to improve the consistency of the filter in scenarios with low or no process noise, while still improving grid placement. In Section IV, the predictive distribution in the FMF is approximated as a Gaussian mixture. In this approximation, each mean is centered at each of the propagated points, all with equal covariance, which is set to the process noise covariance. This section presents a different approach to approximate the predictive distribution, using kernel density estimation (KDE) techniques as used in the EnGMF [10]–[13].

KDE involves estimating the pdf of a state from measurements, by replacing the weight function in a naive estimator with a kernel function $\mathcal{K}$ parameterized by the width β^2. The kernel function has to satisfy certain conditions such as being normalized with respect to the support of the state [19]. Commonly used kernels include the Gaussian distribution, as it maximizes entropy over

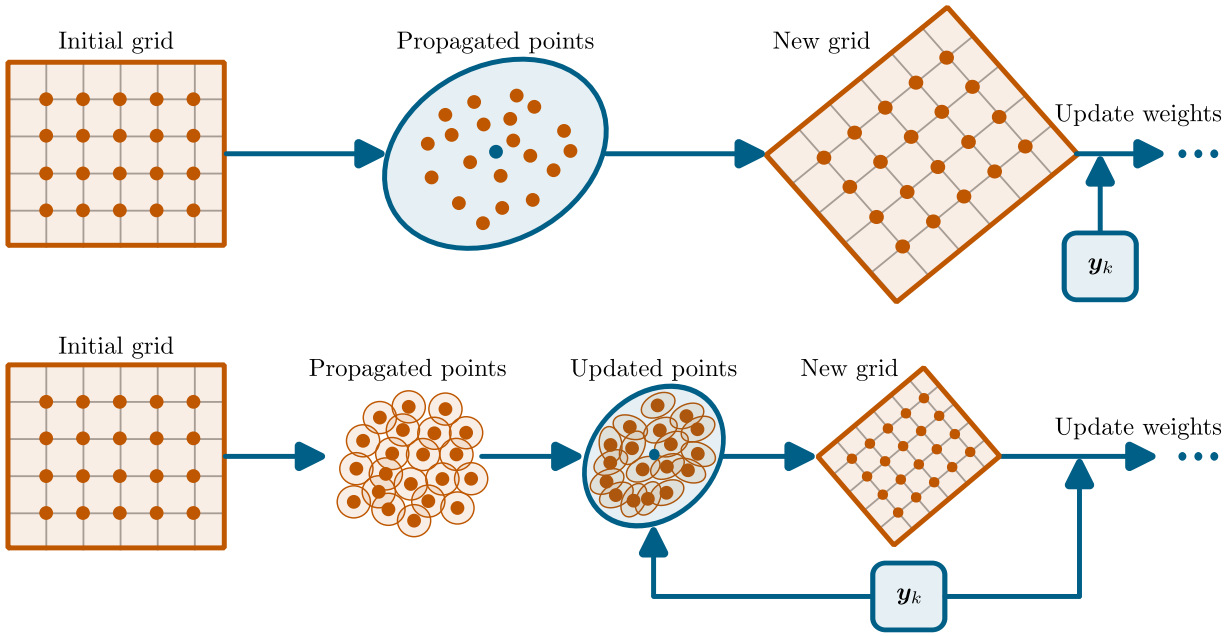

Fig. 2. Comparative analysis between the generation of the predictive grid in the PMF (top) and in the FMF (bottom). The FMF updates the propagated points via a GSF update before creating the new grid. Once the predictive grid has been assembled, the grid points are weighted to create the filtering grid.

all distributions with a given mean and covariance [20], [21].

Assuming that the predictive distribution is unknown, in the standard EnGMF [22], this distribution is calculated as a Gaussian mixture with the means corresponding to the ensemble points and the covariances chosen to be the same for all components of the Gaussian mixture, similar to the FMF approach. The distinction lies in the fact that the covariance is calculated by scaling the sample covariance of the propagated points. Therefore, instead of approximating the predictive distribution as in (22), in this new approach, this distribution is approximated as

$$p\left(\boldsymbol{x}_{k+1}|\boldsymbol{y}_k\right) \approx \sum_{i=1}^{N} w_{k+1|k}^{(i)} \mathcal{N}\left(\boldsymbol{x}_{k+1}; \boldsymbol{f}_k\left(\boldsymbol{\mathcal{X}}_{k|k}^{(i)}\right), B_k\right), \tag{42}$$

where

$$B_k = \beta^2 \hat{P}_{k+1|k} + Q_k, \tag{43}$$

$$\hat{P}_{k+1|k} = \sum_{i=1}^{N} w_{k|k}^{(i)} \left[\boldsymbol{f}_k\left(\boldsymbol{\mathcal{X}}_{k|k}^{(i)}\right) - \hat{\boldsymbol{x}}_{k+1|k}\right] [\cdots]^{\mathrm{T}} \tag{44}$$

$$\hat{\boldsymbol{x}}_{k+1|k} = \sum_{i=1}^{N} w_{k|k}^{(i)} \boldsymbol{f}_k\left(\boldsymbol{\mathcal{X}}_{k|k}^{(i)}\right) \tag{45}$$

where β^2 is the bandwidth parameter, which can be determined heuristically or by prior knowledge of the data being used.

If the predictive distribution is Gaussian and the points are equally weighted, the optimal bandwidth parameter in terms of minimizing the mean integrated squared error is known as Silverman's rule of thumb [19],

$$\beta_{\text{sil}}^2 = \left(\frac{4}{N\left(N_s+2\right)}\right)^{\frac{2}{N_s+4}}, \tag{46}$$

where N_s is the dimension of the state space. Although Silverman's rule of thumb is originally formulated for equally weighted points, it naturally extends to points with unequal weights, as is the case for this filter.

Since the predictive distribution is typically non-Gaussian, Silverman's rule of thumb is known to be overly conservative [23]. In such cases, β_{sil}^2 can be scaled by a factor $\alpha > 0$, to achieve better results in terms of filter consistency,

$$\beta^2 = \alpha \beta_{\text{sil}}^2. \tag{47}$$

As $\alpha \to 0$, the predictive distribution described by (42) converges to the original definition in the FMF as shown in (22). Increasing α will make the approximate distribution of the predictive distribution more conservative, resulting in a more conservative filter. In addition, as $N \to \infty$, $\beta_{\text{sil}}^2 \to 0$, making (22) and (42) equivalent.

The variant of the FMF that uses Silverman's rule of thumb to approximate the predictive distribution is called the SMF. For the SMF, the first weighting scheme described in Section IV is used. Algorithm 1 shows the propagation and update steps for the FMF and SMF. [2] Note that the SMF is an extension of the FMF, as both use the GSF update to improve grid placement. The key difference is that the SMF uses KDE techniques to better approximate the covariance of each point in the Gaussian mixture approximation of the predictive distribution.

VI. ILLUSTRATION WITH A BIVARIATE EXAMPLE

To illustrate the potential improvement in the construction of the FMF and SMF grids, a two-dimensional

[2]MATLAB source code: https://github.com/fgiraldog/GSF-PMF

Algorithm 1 Fusion Silverman mass filter

Input: $\boldsymbol{\mathcal{X}}_{k-1|k-1}$,$\boldsymbol{w}_{k-1|k-1}$,$Q_{k-1}$,$\boldsymbol{y}_k$, R_k, $\alpha, \beta_{\text{sil}}$
Output: $\boldsymbol{\mathcal{X}}_{k|k}, \boldsymbol{w}_{k|k}$

▷ *Propagate the grid:* ◁
for $i = 1 : N$ **do**
  $\boldsymbol{\mathcal{X}}^{(i)}_{k|k-1} = f_{k-1}\left(\boldsymbol{\mathcal{X}}^{(i)}_{k-1|k-1}\right)$
  $\boldsymbol{w}^{(i)}_{k|k-1} = \boldsymbol{w}^{(i)}_{k-1|k-1}$
▷ *Calculate covariance for predictive distribution:* ◁
$P^{(i)}_{k|k-1} = Q_{k-1}$
$P^{(i)}_{k|k-1} = \alpha\beta^2_{\text{sil}} \cdot \hat{P}_{k|k-1} + Q_{k-1}$
▷ *Perform GSF update:* ◁
for $i = 1 : N$ **do**
  $\boldsymbol{\nu}^{(i)}_k = \boldsymbol{y}_k - \boldsymbol{h}_k\left(\boldsymbol{\mathcal{X}}^{(i)}_{k|k-1}\right)$
  $W^{(i)}_k = H^{(i)}_k P^{(i)}_{k|k-1} H^{(i)\mathrm{T}}_k + R_k$
  $K^{(i)}_k = P^{(i)}_{k|k-1} H^{(i)\mathrm{T}}_k \left(W^{(i)}_k\right)^{-1}$
  $\tilde{\boldsymbol{\mathcal{X}}}^{(i)}_{k|k} = \boldsymbol{\mathcal{X}}^{(i)}_{k|k-1} + K^{(i)}_k \boldsymbol{\nu}^{(i)}_k$
  $\tilde{\mathcal{P}}^{(i)}_{k|k} = P^{(i)}_{k|k-1} - K^{(i)}_k W^{(i)}_k \left(K^{(i)}_k\right)^{\mathrm{T}}$
  $\tilde{w}^{(i)}_{k|k} \propto w^{(i)}_{k|k-1} \mathcal{N}\left(\boldsymbol{y}_k; \boldsymbol{h}_k\left(\boldsymbol{x}^{(i)}_{k|k-1}\right), W^{(i)}_k\right)$
▷ *Obtain moments from the updated points:* ◁
$\tilde{\boldsymbol{x}}_{k|k} = \sum_{i=1}^{N} \tilde{w}^{(i)}_{k|k} \tilde{\boldsymbol{\mathcal{X}}}^{(i)}_{k|k}$
$\tilde{P}_{k|k} = \sum_{i=1}^{N} \tilde{w}^{(i)}_{k|k} \left(\tilde{\mathcal{P}}^{(i)}_{k|k} + \tilde{\boldsymbol{\mathcal{X}}}^{(i)}_{k|k}\left(\tilde{\boldsymbol{\mathcal{X}}}^{(i)}_{k|k}\right)^{\mathrm{T}} - \tilde{\boldsymbol{x}}_{k|k}\tilde{\boldsymbol{x}}^{\mathrm{T}}_{k|k}\right)$
▷ *Generate new predictive grid:* ◁
Generate the new grid $\boldsymbol{\mathcal{X}}_k$ from $\left(\tilde{\boldsymbol{x}}_{k|k}, \tilde{P}_{k|k}\right)$
▷ *Weight new grid points (FMF1/SMF):* ◁
$w^{(j)}_{k|k} \propto \sum_{i=1}^{N} \mathcal{N}\left(\boldsymbol{y}_k; \boldsymbol{h}_k\left(\boldsymbol{\mathcal{X}}^{(j)}_k\right), R_k\right) \mathcal{N}\left(\boldsymbol{\mathcal{X}}^{(j)}_k; \boldsymbol{\mathcal{X}}^{(i)}_{k|k-1}, P^{(i)}_{k|k-1}\right) w^{(i)}_{k-1|k-1}$
▷ *Weight new grid points (FMF2):* ◁
$w^{(j)}_{k|k} \propto \sum_{i=1}^{N} \tilde{w}^{(i)}_{k|k} \mathcal{N}\left(\boldsymbol{\mathcal{X}}^{(j)}_k; \tilde{\boldsymbol{\mathcal{X}}}^{(i)}_{k|k}, \tilde{\mathcal{P}}^{(i)}_{k|k}\right)$
▷ *Set weighted grid points as posterior points:* ◁
$\boldsymbol{\mathcal{X}}^{(j)}_{k|k} = \boldsymbol{\mathcal{X}}^{(j)}_k$

problem is presented [24]. The problem starts from a Gaussian predictive distribution,

$$p(\boldsymbol{x}_{k+1}|\boldsymbol{y}_k) \sim \mathcal{N}\left(\begin{bmatrix} -3.5 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}\right), \tag{48}$$

from which 625 equally weighted realizations are sampled, simulating the propagation of an initial posterior grid.

Using the mean and covariance of the predictive distribution, a uniform predictive grid is created for the standard PMF. The new points of the predictive grid are convoluted with the equally weighted realizations using additive white Gaussian process noise with a covariance matrix of $Q = 2 \times 10^{-1} I_{2\times 2}$. The points on the predictive grid are then weighted to generate the filtering grid, according to the following measurement,

$$y = h(\boldsymbol{x}) = \sqrt{\left(x^{(1)}\right)^2 + \left(x^{(2)}\right)^2}, \tag{49}$$

with a value of $y = 1$ and associated scalar measurement covariance of $R = 0.1^2$.

For the FMF and SMF, the equally weighted realizations are updated with the GSF formulation by using the same measurement, measurement covariance, and process noise covariance matrices as for the PMF. A new uniform grid is created centered on the updated mean and scaled by the updated covariance. The new points on the grid are then weighted as in (40), generating the filtering grid.

For comparison, three alternative approaches for improving the grid position in the PMF are presented, using ideas from previous work that align with the current framework.

1) The first approach, PMF-BER, is motivated by the work of Bergman [5], [6]. For the implementation presented in this work, a posterior mean and covariance are calculated from the PMF filtering grid. Then a new uniform grid is centered on the new mean, scaled by the new covariance, and weighted as described in (40).
2) The second approach, PMF-UKF, uses some of the ideas presented by Duník et al. [7]. Instead of using a GSF update, a UKF update is applied to obtain a posterior mean and covariance. Once these estimates are obtained, a uniform grid is centered on the updated mean, scaled by the updated covariance, and weighted as described in (40).
3) The third approach, PMF-DHF, is based on the work of Choe and Park [8]. The implementation presented here flows the boundary of the standard PMF grid using the exact Daum and Huang flow [25] with n_λ pseudo-time steps. From the flowed boundary, a new grid support is determined using a minimum bounding box algorithm [26], which identifies the arbitrarily oriented minimum-area rectangle that contains all the flowed grid points. Then a uniform grid is placed on the new grid support and is weighted as in (40).

For all approaches, the number of grid points remains constant, i.e. $N = 625$. All grids, except for the PMF-DHF which uses the minimum bounding box algorithm, are scaled to 3σ of their respective covariance matrices using a similar implementation as in [27].

Figure 3 shows the filtering grids for the six approaches. From this figure, it can be seen that the standard PMF generates a predictive grid centered on the prior distribution, resulting in a filtering grid with few points near the true posterior. The UKF linear update in the PMF-UKF is inaccurate in this example, resulting in suboptimal grid placement. However, the PMF-UKF grid is closer to the posterior than the grid in the standard PMF. A similar behavior is observed with the PMF-DHF, where the grid is also closer to the posterior but still struggles to capture the lower tail of the distribution.

In comparison, the PMF-BER, FMF, and SMF generate a filtering grid that more accurately approximates the true posterior than the other PMF variants. Note that

the PMF-BER grid is constrained by the standard PMF grid. If the standard PMF grid is far from the posterior, the PMF-BER grid will not capture the posterior well. This limitation is addressed in the FMF and SMF by incorporating the GSF update before grid creation, which, for this example, centers the filtering grid on the posterior. Comparing the FMF and SMF, the advantages of using Silverman's rule of thumb can be seen. While the FMF represents an improvement over the other PMF variants, it can be observed that the grid may also struggle to capture the end portion of the lower tail in the posterior distribution. The SMF has a more conservative approach, in which the grid covers a larger area that is capable of capturing the entire posterior distribution.

VII. SEQUENTIAL FILTERING WITH IKEDA MAP

The following example aims to demonstrate the performance improvement of the FMF and SMF by conducting a grid study in a sequential filtering problem. To this end, the Ikeda map is used, representing a discrete-time dynamical system,

$$x_{k+1}^{(1)} = 1 + u\left(x_k^{(1)}\cos t_k - x_k^{(2)}\sin t_k\right), \tag{50}$$

$$x_{k+1}^{(2)} = u\left(x_k^{(1)}\sin t_k + x_k^{(2)}\cos t_k\right), \tag{51}$$

with,

$$t_k = 0.4 - \frac{6}{1 + \left(x_k^{(1)}\right)^2 + \left(x_k^{(2)}\right)^2}. \tag{52}$$

This system is designed to model light circulating within a nonlinear optical resonator [28], [29]. To introduce chaos and create a more challenging filtering problem, $u = 0.9$, and the dynamics are propagated with additive white Gaussian process noise with covariance matrix $Q = 1 \times 10^{-2} I_{2\times 2}$. A nonlinear measurement model is assumed, with measurements given by

$$y_k = \sqrt{\left(x_k^{(1)}\right)^2 + \left(x_k^{(2)}\right)^2} + \eta_k, \tag{53}$$

where η_k represents white Gaussian measurement noise with scalar covariance matrix $R = 1$.

Each of the previously defined filters is tested on 1,000 distinct trajectories, each simulated for 50 time steps, starting from an initial true state defined as $\boldsymbol{x}_0 \sim \mathcal{N}(0_{2\times 1}, I_{2\times 2})$. The performance of each filter is evaluated by varying the number of grid points used. In each test, the grids for all filters, except PMF-DHF which uses the minimum bounding box algorithm, are scaled to 3σ of their respective covariance matrices.

To compare each filter, two metrics are used, the time-averaged root mean squared error (RMSE) and the time-averaged scaled normalized estimation error squared (SNEES). The RMSE is used to assess the accuracy of

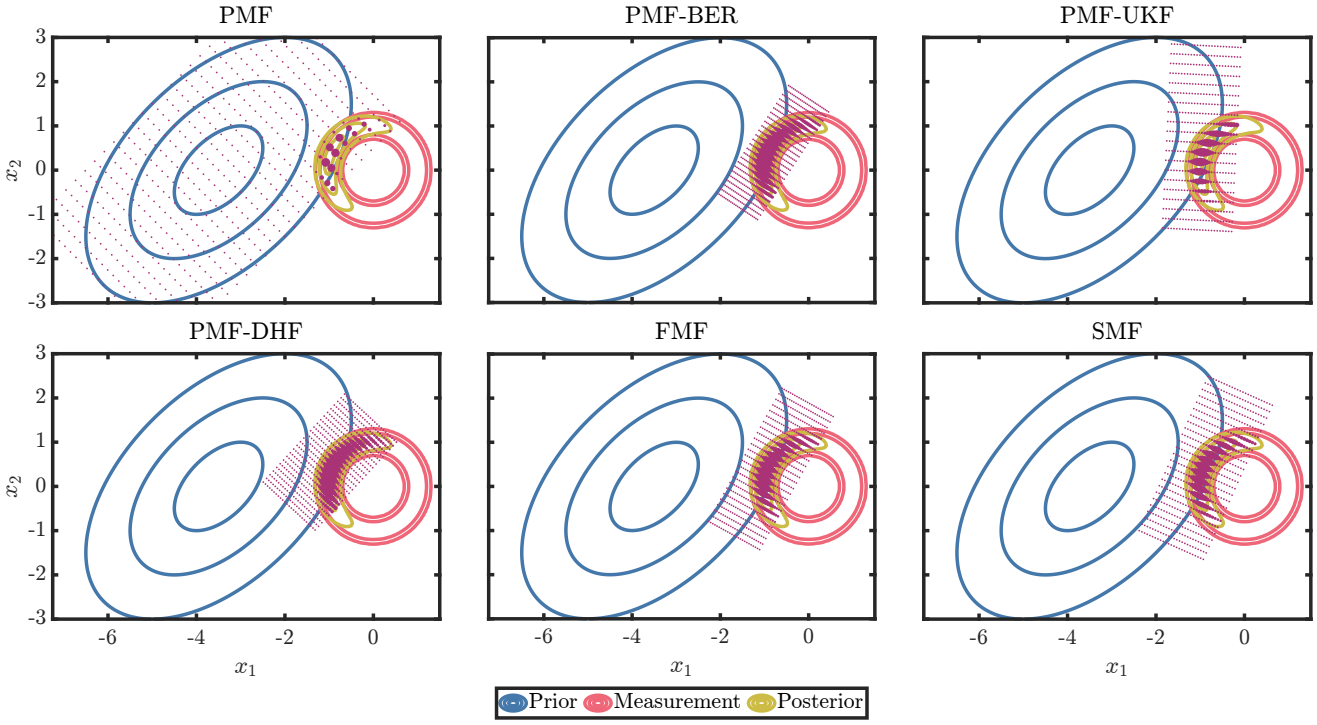

Fig. 3. Comparison between the PMF (top left), PMF-BER (top center), PMF-UKF (top right), PMF-DHF (bottom left) with $n_\lambda = 25$, FMF (bottom center), and SMF (bottom right) with $\alpha = 1$. The prior distribution is shown in blue, with contours at 1σ, 2σ, and 3σ. The measurement likelihood is plotted in red, with contours at 2σ and 3σ, while the true posterior distribution is shown in yellow, with contours at 1σ, 2σ, and 3σ. The filtering grids are shown in purple, with each grid point plotted based on its weight. Larger points indicate higher weights, while smaller points carry lower weights.

the filter defined as

$$\text{RMSE} = \frac{1}{N_t N_m} \sum_{k=1}^{N_t} \sum_{j=1}^{N_m} \sqrt{\sum_{i=1}^{N_s} \frac{\left(x_{k,j}^{(i)} - \hat{x}_{k|k,j}^{(i)}\right)^2}{N_s}}, \quad (54)$$

where N_t is the total number of discrete steps in each simulation, N_m is the number of independent trajectories, N_s is the state dimension, $x_{k,j}^{(i)}$ is the true state and $\hat{x}_{k|k,j}^{(i)}$ is the estimated state. The SNEES is used to evaluate the consistency of the filter defined as

$$\text{SNEES} = \frac{1}{N_t} \sum_{k=1}^{N_t} \sum_{j=1}^{N_m} \frac{\boldsymbol{e}_{k,j}^{\mathrm{T}} \left(P_{k|k,j}\right)^{-1} \boldsymbol{e}_{k,j}}{N_m N_s}, \quad (55)$$

where

$$\boldsymbol{e}_{k,j} = \boldsymbol{x}_{k,j} - \hat{\boldsymbol{x}}_{k|k,j}, \quad (56)$$

is the estimation error and $P_{k|k,j}$ is the estimated covariance. A consistent filter will obtain a SNEES close to one. If the SNEES value exceeds one, it suggests overconfidence in the estimator. Conversely, if the value is less than one, the estimator is conservative [1]. It is important to note that for this section, any SNEES value greater than 1×10^4 was considered a numerical instability and was therefore disregarded.

Figure 4 shows the RMSE for each approach as a function of the number of points on the grid. For the SMF, Silverman's rule of thumb is scaled by $\alpha = 0.2$. For the PMF-DHF, $n_\lambda = 25$. As expected, increasing the number of grid points improves the accuracy for all filters. The PMF-BER and PMF-UKF show a similar accuracy, whereas the PMF-DHF performs slightly better than the standard PMF in grids with a low number of points. Note that the exact flow approximates the prior distribution as a Gaussian [30], which is not the case for this example, potentially explaining the small improvement between the PMF-DHF and the standard PMF. Across all grid sizes, the FMF1, FMF2, and SMF consistently outperform the other PMF variants. The FMF1 and FMF2 yield similar results, indicating that both weighting schemes deliver comparable performance. The SMF performs significantly well compared to the other filters, being capable of handling low process noise scenarios driven by nonlinear dynamics with fewer grid points.

Figure 5 shows the SNEES for all filters as a function of the number of points on the grid. Similarly to the RMSE figure, all filters show increasing consistency with the use of more grid points. In this case, the PMF-BER and PMF-UKF show similar consistency, while the PMF-DHF follows the standard PMF. Both the FMF1 and FMF2 approach a SNEES close to one around 300 grid points, outperforming the other PMF variants. The SMF achieves a SNEES value of one with a small number of grid points, highlighting the benefits of using Silverman's rule of thumb and explaining the difference in accuracy results shown in Fig 4.

Considering the different techniques used for each filter, it is important to quantify the computational runtime of these algorithms. Figure 6 shows the RMSE of each

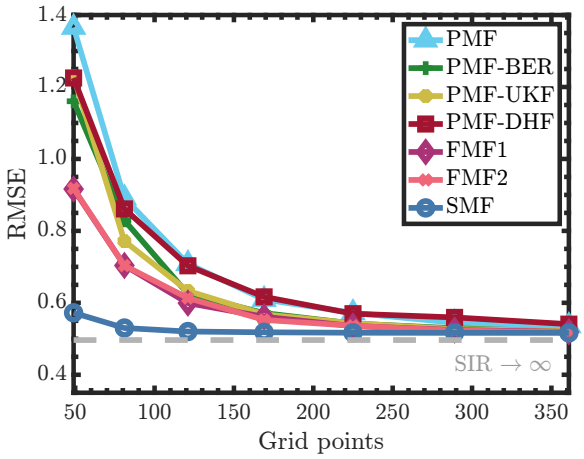

Fig. 4. RMSE for the different PMF variants as a function of grid points. The gray line represents the results obtained with a regularized particle filter using 10,000 particles and serves as a reference.

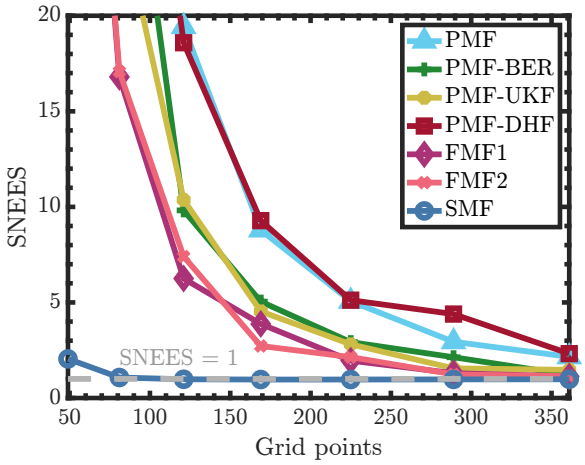

Fig. 5. SNEES for the different PMF variants as a function of grid points. The gray dashed line represents a SNEES value of one.

filter as a function of their required runtime for a single propagation and update. All filters are optimized to maximize runtime efficiency for fair comparisons. Specifically, the convolution step is vectorized for all filters. The GSF update and the boundary flow step are also performed in a vectorized fashion. The filters were run on a MacBook Pro with an Apple M1 Pro chip and 16 GB of RAM, implemented in MATLAB R2024a.

From Fig. 6, it can be seen that the FMF2 is considerably slower than the FMF1. The main reason for this slowdown is the weighting scheme used in this filter (41), since for each point on the grid, a different covariance matrix has to be inverted. For the FMF1 and SMF, only one covariance matrix, namely Q_k or B_k, has to be inverted for all grid points. The PMF-BER nearly doubles the standard PMF runtime as the convolution step is performed twice. The PMF-DHF is slower than other variants (except FMF2) due to its need to flow the boundary. Comparing the PMF-UKF, FMF1, and SMF with the PMF, the extra step in these filters is noticeable by the slight shift to the right in their curves. However,

as shown before, fewer grid points are needed for both the FMF1 and SMF to achieve more accurate results compared to the other PMF variants. The use of the GSF update and Silverman's rule of thumb allows the implementation of smaller grids, which translates into faster runtime accompanied by a low RMSE. Table I shows the final ($N = 361$) RMSE, SNEES, and runtime values for all filters tested.

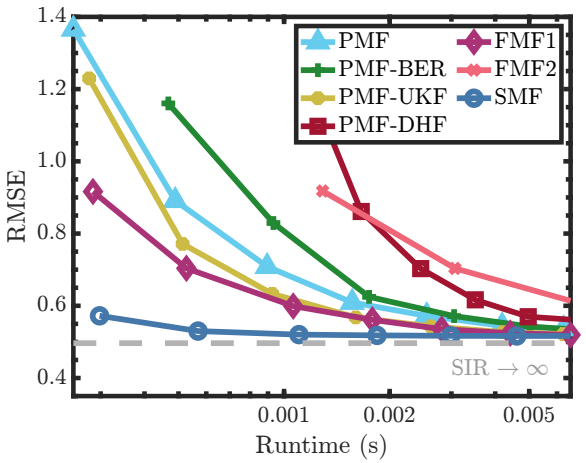

Fig. 6. RMSE for the different PMF variants as a function of runtime for one propagation and update. The gray line represents the results obtained with a regularized particle filter using 10,000 particles and serves as a reference.

TABLE I
RMSE, SNEES, and runtime obtained for each filter presented ($N = 361$)

Filter	RMSE	SNEES	Runtime (s)
PMF	0.5359	2.1549	0.0063
PMF-BER	0.5199	1.2110	0.0123
PMF-UKF	0.5217	1.4732	0.0063
PMF-DHF	0.5407	2.3212	0.0099
FMF1	0.5194	1.1277	0.0066
FMF2	0.5183	1.1422	0.0637
SMF	0.5165	0.9806	0.0069

VIII. INTERLUDE: A CLOSER LOOK AT THE SMF

This section analyzes the effect of α on the SMF and compares this filter with previous works that have addressed the low process noise issue in the PMF. The Ikeda map is used again with the same settings as in Section VII.

Since the SMF depends on α, its impact on filtering accuracy and consistency has to be evaluated. As $\alpha \to 0$, the predictive distribution described by (42) converges to the original definition in the FMF. Increasing α will make the approximation of the predictive distribution more conservative, leading to a more conservative filter. This behavior can be seen in Fig. 7, where the RMSE and SNEES for the SMF are presented as a function of α for a fixed number of grid points, $N = 361$. For each α, 1,000 trajectories are simulated for 50 time steps. In the case of $\alpha = 0$, the RMSE and SNEES of the SMF converge to the FMF. As α increases, the consistency of the SMF improves, approaching a SNEES of one. The RMSE reaches its minimum when the SNEES value for the SMF crosses one. Once α is relatively large and the SNEES value starts to drop below one, the accuracy of the filter degrades as expected. It should be noted that α can be determined algorithmically [31].

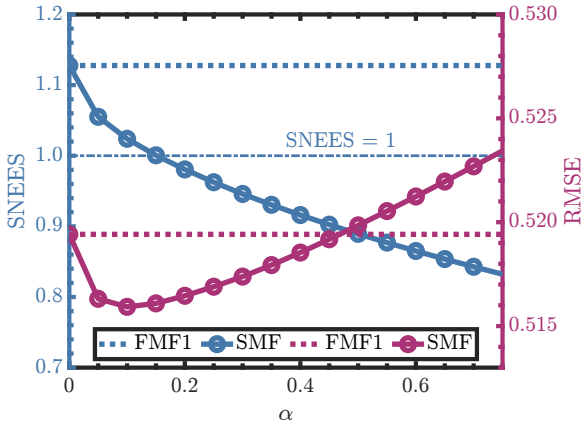

Fig. 7. SNEES (left) for the FMF1 (dashed line) and SMF (solid line) as a function of α. RMSE (right) for the FMF1 (dashed line) and SMF (solid line) as a function of α.

As mentioned in Section V, Duník et al. [15] proposed addressing the issue of low process noise in the PMF using a density-weighted convolution (DWC). For the DWC, the Dirac mixture approximation is dropped, and the pdfs are approximated as a piecewise function. The proposed solution then solves, analytically or numerically, the convolution integral in (6) based on the new functional form of the approximated pdf. Note that a key difference between this solution and the SMF is that it addresses only the convolution step, as it uses the standard PMF grid. As a result, the grid placement is not improved as it is in the SMF.

In this work, the SMF is compared to the solution proposed in [15] by implementing the DWC in two different ways:

1) The first approach, PMF-DWC$_a$, solves the convolution integral analytically by linearizing the dynamics and using the `mvncdf` MATLAB function.
2) The second approach, PMF-DWC$_n$, solves the convolution integral numerically by using the `integral2` MATLAB function.

To compare the PMF-DWC variants with the SMF, the three filters are evaluated based on RMSE, SNEES, and runtime, using a fixed number of grid points. The number of grid points is set to $N = 49$, representing the lowest number tested in Figs. 4 and 5, and the grids are scaled to 3σ of their respective covariance matrices.

In the case of the SMF, $\alpha = 0.4$ is used for a more conservative approach. Additionally, the convolution in the SMF is performed without vectorization because the `integral2` MATLAB function is not easily vectorized. This adjustment ensures that all filters have comparable implementations, resulting in valid runtime comparisons. Table II shows the RMSE, SNEES, and runtime for the PMF-DWC$_{a,n}$ and the SMF obtained by simulating 1,000 trajectories for 50 time steps.

TABLE II
RMSE, SNEES, and runtime obtained for the SMF and the PMF-DWC variants ($N = 49$)

Filter	RMSE	SNEES	Runtime (s)
PMF-DWC$_a$	0.6033	3.5771	0.1375
PMF-DWC$_n$	0.5585	2.1763	0.4385
SMF	0.5418	1.0274	0.0011

From Table II, it can be seen that performing the DWC in the standard PMF significantly improves both its consistency and accuracy. However, both PMF-DWC variants are two orders of magnitude more computationally expensive than the SMF, which uses the standard convolution. This aligns with the results in [15], where the DWC was shown to be more computationally expensive than the standard convolution. Note that some techniques can be used to reduce the computational complexity of the DWC [15]. For the PMF-DWC$_a$, linearizing the dynamics slightly affects the performance due to the chaotic behavior of the Ikeda map. Although the PMF-DWC$_n$ obtains results closer to the SMF, the SMF remains more accurate and consistent, as it also improves grid placement.

IX. MARS TERRAIN-RELATIVE NAVIGATION

The PMF is a common approach for terrain-relative navigation (TRN) [5], [7], [32]. In this type of problem, an elevation map is used onboard to estimate the horizontal position of a vehicle based on measurements from a barometric sensor and an altimeter [5], [7]. To show the performance of the FMF and SMF in a more practical setup, a grid study is presented for a Mars TRN problem. In addition, this section presents a new estimation scheme similar to Rao-Blackwellized solutions [32], [33], by using the Q-method EKF update [14]. The Q-method EKF update is useful when measurements depend only on specific states. This technique allows nonlinear filters, such as PMFs, to update the states in the measurement model, while the remaining states are updated linearly, as in a standard EKF.

In this example, the horizontal position of a Mars reconnaissance vehicle is estimated via the filters previously presented, while the velocity of the vehicle is estimated linearly. This estimation scheme reduces the computational cost of estimating both position and velocity via the use of nonlinear filters. The problem setup

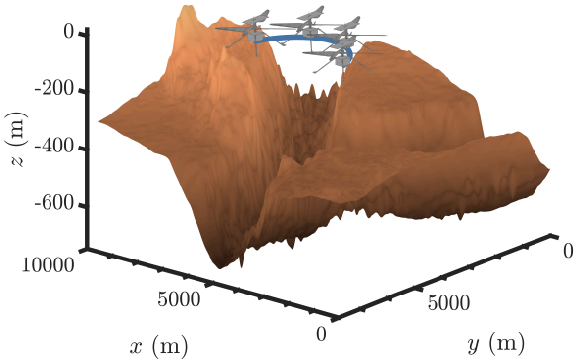

Fig. 8. Digital elevation map of the Nili Fossae, located north of Jezero Crater. The simulated nominal trajectory is shown in blue.

simulates a constant-altitude and constant-turn flight with a known turn rate over the Nili Fossae, located north of Jezero Crater. This scenario is reminiscent of possible reconnaissance flights on Mars by small helicopters, such as Ingenuity, but with a longer flying range. The discrete dynamics of the trajectory are given by [34], [35]:

$$\begin{bmatrix} r_{k+1}^{(1)} \\ r_{k+1}^{(2)} \\ v_{k+1}^{(1)} \\ v_{k+1}^{(2)} \end{bmatrix} = F \begin{bmatrix} r_{k}^{(1)} \\ r_{k}^{(2)} \\ v_{k}^{(1)} \\ v_{k}^{(2)} \end{bmatrix} + \boldsymbol{q}_k, \tag{57}$$

$$F = \begin{bmatrix} 1 & 0 & \frac{\sin(\omega\Delta t_k)}{\omega} & -\frac{(1-\cos(\omega\Delta t_k))}{\omega} \\ 0 & 1 & \frac{(1-\cos(\omega\Delta t_k))}{\omega} & \frac{\sin(\omega\Delta t_k)}{\omega} \\ 0 & 0 & \cos(\omega\Delta t_k) & -\sin(\omega\Delta t_k) \\ 0 & 0 & \sin(\omega\Delta t_k) & \cos(\omega\Delta t_k) \end{bmatrix}, \tag{58}$$

where $r^{(i)}$ and $v^{(i)}$ are the horizontal position and velocity in the i-th direction respectively, Δt_k is the time-step, ω is the known turn rate and $\boldsymbol{q}_k$ is white Gaussian noise with covariance matrix [34], [35],

$$Q_k = S_{\boldsymbol{q}} \begin{bmatrix} \frac{2\omega_{\sin}}{\omega^3} & 0 & \frac{\omega_{\cos}}{\omega^2} & \frac{\omega_{\sin}}{\omega^2} \\ 0 & \frac{2\omega_{\sin}}{\omega^3} & -\frac{\omega_{\sin}}{\omega^2} & \frac{\omega_{\cos}}{\omega^2} \\ \frac{\omega_{\cos}}{\omega^2} & -\frac{\omega_{\sin}}{\omega^2} & \Delta t_k & 0 \\ \frac{\omega_{\sin}}{\omega^2} & \frac{\omega_{\cos}}{\omega^2} & 0 & \Delta t_k \end{bmatrix}, \tag{59}$$

$$\omega_{\sin} = \omega\Delta t_k - \sin(\omega\Delta t_k), \tag{60}$$

$$\omega_{\cos} = 1 - \cos(\omega\Delta t_k), \tag{61}$$

where $S_{\boldsymbol{q}} I_{2\times 2}$ is the power spectral density of the white noise. For this work, the constant turn is performed with a known turn rate of $\omega = 1.2 \times 10^{-2}$ rad/s, a time step of $\Delta t_k = 1$ second, over a total duration of 200 seconds. The trajectory is corrupted with white noise with $S_{\boldsymbol{q}} = 1 \times 10^{-4}$ m^2/s^3.

Figure 8 shows the digital elevation map[3], generated from the Mars Reconnaissance Orbiter (MRO) Context Camera (CTX) [36], and the simulated nominal trajectory starting from an initial state defined as,

$$\boldsymbol{x}_0 = \begin{bmatrix} 3400 \text{ m} & 3500 \text{ m} & 8 \ \frac{\text{m}}{\text{s}} & -21 \ \frac{\text{m}}{\text{s}} \end{bmatrix}^{\text{T}}. \tag{62}$$

The measurement used for this problem is the elevation of the terrain given by the map at the current position, simulating possible measurements from a barometric sensor and an altimeter. As the map provides elevations at discrete positions, cubic spline interpolation is used to obtain the elevation at the query positions. Each measurement is corrupted with additive white Gaussian noise with a scalar covariance matrix $R = 0.1^2$. Figure 9 shows the noiseless elevation profile of the nominal trajectory.

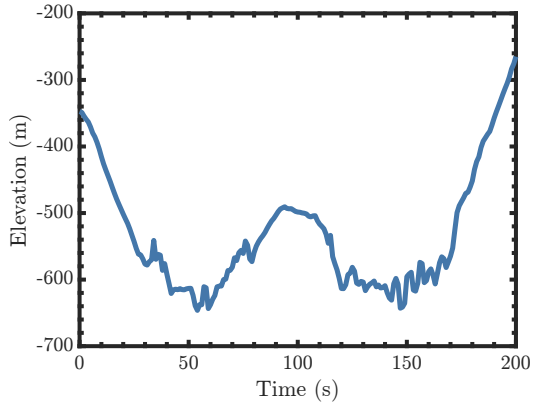

Fig. 9. Terrain elevation profile of the nominal trajectory.

The described PMF variants are used to estimate the horizontal position using the following procedure. Starting with the current posterior position grid ($\boldsymbol{\mathcal{R}}_{k|k}$), this grid is propagated using a uniform velocity grid ($\boldsymbol{\mathcal{V}}_{k|k}$) centered on the posterior velocity estimate ($\hat{\boldsymbol{v}}_{k|k}$) and scaled by its corresponding covariance $\left(P^{(\boldsymbol{v},\boldsymbol{v})}_{k|k}\right)$, such that

$$\begin{bmatrix} \boldsymbol{\mathcal{R}}^{(i)}_{k+1|k} \\ \boldsymbol{\mathcal{V}}^{(i)}_{k+1|k} \end{bmatrix} = \begin{bmatrix} F^{(\boldsymbol{r},\boldsymbol{r})} & F^{(\boldsymbol{r},\boldsymbol{v})} \\ \mathbf{0} & F^{(\boldsymbol{v},\boldsymbol{v})} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mathcal{R}}^{(i)}_{k|k} \\ \boldsymbol{\mathcal{V}}^{(i)}_{k|k} \end{bmatrix}, \tag{63}$$

where $\boldsymbol{\mathcal{R}}^{(i)}_{k|k}$ and $\boldsymbol{\mathcal{V}}^{(i)}_{k|k}$ are the position and velocity of the i-th grid point, respectively. The position grid is propagated using a velocity grid to ensure that the resulting grid captures the correlation between position and velocity. Furthermore, since the dynamics are linear, the propagated velocity estimate ($\hat{\boldsymbol{v}}_{k+1|k}$) is equivalent to the mean of the propagated velocity grid. Once an elevation measurement is obtained ($k \leftarrow k+1$) a new position grid is constructed and weighted using any of the PMF variants discussed. It is important to note that, since the PMF variants do not directly estimate velocity, the covariance matrix of the transitional pdf has to account for the uncertainty in velocity, where

$$Q^{\text{Total}}_{k-1} = F^{(\boldsymbol{r},\boldsymbol{v})} P^{(\boldsymbol{v},\boldsymbol{v})}_{k-1|k-1} F^{(\boldsymbol{r},\boldsymbol{v})\text{T}} + Q^{(\boldsymbol{r},\boldsymbol{r})}_{k-1}. \tag{64}$$

After the horizontal position has been estimated, the Q-method EKF update [14] is used to estimate the velocity. First, an analytic covariance is propagated using the EKF formulation,

$$\bar{P}_{k|k-1} = F P_{k-1|k-1} F^{\text{T}} + Q_k. \tag{65}$$

With the propagated covariance, the velocity estimates are updated as follows [14],

$$\hat{\boldsymbol{v}}_{k|k} = \hat{\boldsymbol{v}}_{k|k-1} + \bar{P}^{(\boldsymbol{v},\boldsymbol{r})}_{k|k-1} \bar{P}^{(\boldsymbol{r},\boldsymbol{r})-1}_{k|k-1} \left(\hat{\boldsymbol{r}}_{k|k} - \hat{\boldsymbol{r}}_{k|k-1}\right), \tag{66}$$

$$P^{(\boldsymbol{v},\boldsymbol{r})}_{k|k} = \bar{P}^{(\boldsymbol{v},\boldsymbol{r})}_{k|k-1} \bar{P}^{(\boldsymbol{r},\boldsymbol{r})-1}_{k|k-1} P^{(\boldsymbol{r},\boldsymbol{r})}_{k|k}, \tag{67}$$

$$P^{(\boldsymbol{v},\boldsymbol{v})}_{k|k} = \bar{P}^{(\boldsymbol{v},\boldsymbol{v})}_{k|k-1} + \bar{P}^{(\boldsymbol{v},\boldsymbol{r})}_{k|k-1} \Big(\bar{P}^{(\boldsymbol{r},\boldsymbol{r})-1}_{k|k-1} P^{(\boldsymbol{r},\boldsymbol{r})}_{k|k} \bar{P}^{(\boldsymbol{r},\boldsymbol{r})-1}_{k|k-1} - \bar{P}^{(\boldsymbol{r},\boldsymbol{r})-1}_{k|k-1} \Big) P^{(\boldsymbol{r},\boldsymbol{v})}_{k|k}, \tag{68}$$

where $\hat{\boldsymbol{r}}_{k|k-1}$ is the propagated position estimate obtained from the propagated position grid, and $\hat{\boldsymbol{r}}_{k|k}$ and $P^{(\boldsymbol{r},\boldsymbol{r})}_{k|k}$ are the posterior position estimate and its corresponding covariance obtained from the filtering position grid. Algorithm 2 provides a summary of the propagation and update steps for this specific problem. Figure 10 presents an overview of the online estimation scheme.

For every filter tested, all grids are constructed up to 5σ of their respective covariance matrices, to adopt a more conservative approach. Each filter is evaluated in 1,000 trajectories, with an initial true state distributed as a Gaussian distribution with mean given by the initial state in (62) and covariance,

$$P_0 = \begin{bmatrix} 50 I_{2\times 2} \text{ m}^2 & 0_{2\times 2} \ \frac{\text{m}^2}{\text{s}} \\ 0_{2\times 2} \ \frac{\text{m}^2}{\text{s}} & I_{2\times 2} \ \left(\frac{\text{m}}{\text{s}}\right)^2 \end{bmatrix}. \tag{69}$$

For this example, the FMF2 is not presented as it yields similar results to FMF1. The PMF-DHF is also not included because, as implemented, it is not compatible with the problem framework. Since the PMF-DHF grid support is defined by the arbitrarily oriented minimum bounding box of the flowed grid points, its orientation is not guaranteed to align with the velocity grid used for propagation, resulting in poor performance.

Figure 11 shows the *position* RMSE for the PMF, PMF-BER, PMF-UKF, FMF1, and SMF as a function of the number of grid points. For the SMF, Silverman's rule of thumb is scaled by a factor of $\alpha = 0.1$. Similarly to the results obtained with the Ikeda map, the curve for the FMF1 and SMF remains closer to the origin than the other PMFs, showing an improvement in filtering accuracy. In this case, the PMF-BER outperforms the PMF-UKF, approaching the results obtained with the FMF1. The position RMSE for the EKF is presented for comparison purposes. As expected, the nonlinear filters outperform the EKF solution in terms of accuracy.

[3]Elevation map obtained from The Pacific Regional Planetary Data Center hosted by the Hawai'i Institute of Geophysics and Planetology.

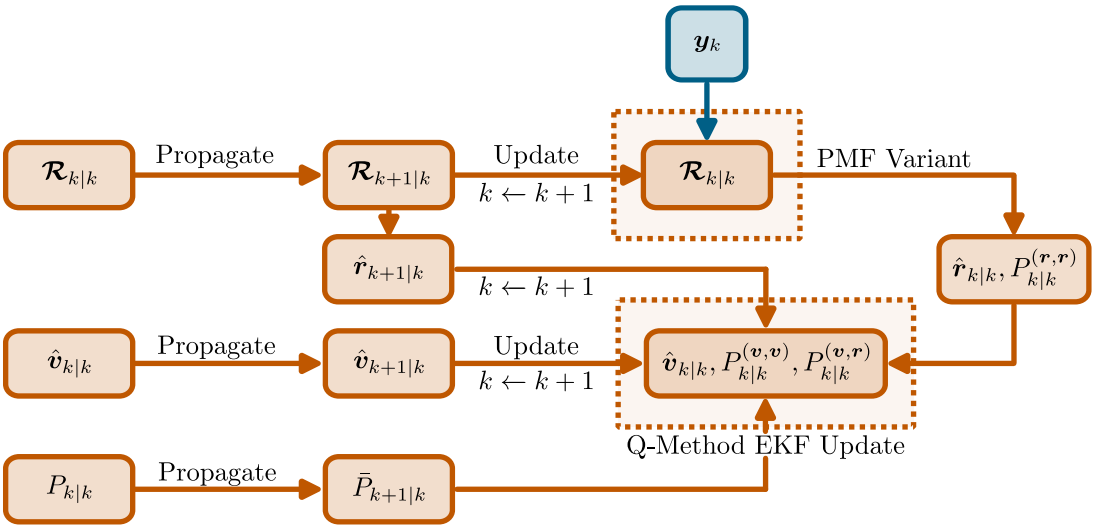

Fig. 10. Flowchart of the terrain relative navigation algorithm. $\mathcal{R}$ represents the position grid, $\boldsymbol{r}$ is the position, $\boldsymbol{v}$ is the velocity, P is the covariance matrix, and $\boldsymbol{y}$ is the measurement.

Algorithm 2 TRN Estimation Algorithm

Input: $\mathcal{R}_{k-1|k-1}, \boldsymbol{w}_{k-1|k-1}, \hat{\boldsymbol{v}}_{k-1|k-1}, P_{k-1|k-1}, Q_{k-1}, \boldsymbol{y}_k, R_k$

Output: $\mathcal{R}_{k|k}, \boldsymbol{w}_{k|k}, \hat{\boldsymbol{v}}_{k|k}, P_{k|k}$

▷ *Build velocity grid:* ◁

$\mathcal{V}_{k-1|k-1} \leftarrow \left(\hat{\boldsymbol{v}}_{k-1|k-1}, P^{(\boldsymbol{v},\boldsymbol{v})}_{k-1|k-1}\right)$

▷ *Propagate the grids:* ◁

for $i = 1 : N$ **do**

$\quad \mathcal{R}^{(i)}_{k|k-1} = F^{(\boldsymbol{r},\boldsymbol{r})}\mathcal{R}^{(i)}_{k-1|k-1} + F^{(\boldsymbol{r},\boldsymbol{v})}\mathcal{V}^{(i)}_{k-1|k-1}$

$\quad \mathcal{V}^{(i)}_{k|k-1} = F^{(\boldsymbol{v},\boldsymbol{v})}\mathcal{V}^{(i)}_{k-1|k-1}$

▷ *Propagate EKF covariance:* ◁

$\bar{P}_{k|k-1} = FP_{k-1|k-1}F^T + Q_k$

▷ *Get prior estimates:* ◁

$\hat{\boldsymbol{r}}_{k|k-1} = \sum_{i=1}^{N} w^{(i)}_{k-1|k-1}\mathcal{R}^{(i)}_{k|k-1}$

$\hat{\boldsymbol{v}}_{k|k-1} = \sum_{i=1}^{N} \mathcal{V}^{(i)}_{k|k-1}/N$

▷ *Calculate total noise matrix:* ◁

$Q^{\text{Total}}_{k-1} = F^{(\boldsymbol{v},\boldsymbol{v})}P^{(\boldsymbol{v},\boldsymbol{v})}_{k-1|k-1}\left(F^{(\boldsymbol{v},\boldsymbol{v})}\right)^{\mathrm{T}} + Q^{(\boldsymbol{r},\boldsymbol{r})}_{k-1}$

▷ *Generate and weight new position grid:* ◁

Generate new weighted position grid $\left(\mathcal{R}_{k|k}, \boldsymbol{w}_{k|k}\right)$ using any of the PMF variants.

▷ *Obtain first two moments from the updated grid:* ◁

$\hat{\boldsymbol{r}}_{k|k} = \sum_{i=1}^{N} w^{(i)}_{k|k}\mathcal{R}^{(i)}_{k|k}$

$P^{(\boldsymbol{r},\boldsymbol{r})}_{k|k} = \sum_{i=1}^{N} w^{(i)}_{k|k}\left[\mathcal{R}^{(i)}_{k|k}\left(\mathcal{R}^{(i)}_{k|k}\right)^{\mathrm{T}}\right] - \hat{\boldsymbol{r}}_{k|k}\hat{\boldsymbol{r}}^{\mathrm{T}}_{k|k}$

▷ *Update velocity estimate* ◁

$\hat{\boldsymbol{v}}_{k|k} = \hat{\boldsymbol{v}}_{k|k-1} + \bar{P}^{(\boldsymbol{v},\boldsymbol{r})}_{k|k-1}\bar{P}^{(\boldsymbol{r},\boldsymbol{r})-1}_{k|k-1}\left(\hat{\boldsymbol{r}}_{k|k} - \hat{\boldsymbol{r}}_{k|k-1}\right)$

▷ *Update remaining covariance terms* ◁

$P^{(\boldsymbol{v},\boldsymbol{r})}_{k|k} = \bar{P}^{(\boldsymbol{v},\boldsymbol{r})}_{k|k-1}\bar{P}^{(\boldsymbol{r},\boldsymbol{r})-1}_{k|k-1}P^{(\boldsymbol{r},\boldsymbol{r})}_{k|k}$

$P^{(\boldsymbol{v},\boldsymbol{v})}_{k|k} = \bar{P}^{(\boldsymbol{v},\boldsymbol{v})}_{k|k-1} + \bar{P}^{(\boldsymbol{v},\boldsymbol{r})}_{k|k-1}\left(\bar{P}^{(\boldsymbol{r},\boldsymbol{r})-1}_{k|k-1}P^{(\boldsymbol{r},\boldsymbol{r})}_{k|k}\bar{P}^{(\boldsymbol{r},\boldsymbol{r})-1}_{k|k-1} - \bar{P}^{(\boldsymbol{r},\boldsymbol{r})-1}_{k|k-1}\right)P^{(\boldsymbol{r},\boldsymbol{v})}_{k|k}$,

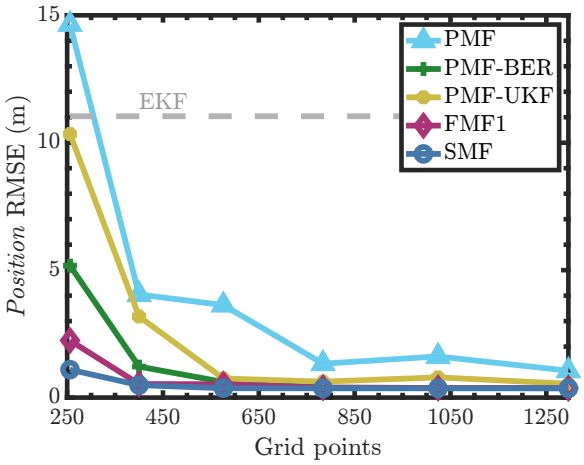

Fig. 11. Position RMSE for the different PMF variants as a function of grid points. The gray line represents the results obtained with an EKF and serves as a reference.

The *velocity* RMSE as a function of the number of grid points can be seen in Fig. 12. Similarly to the position RMSE, a comparable behavior is observed in this figure. Both the FMF1 and SMF perform better than the PMF-UKF, PMF-BER, and PMF. It is important to note that the performance of the PMF-UKF degrades with low number of grid points, as the velocity error tends to be close to that of the PMF. This is due to the linear nature of the UKF update, which can negatively impact the filter if a low number of grid points is used. However, as more grid points are added, the performance of the PMF-UKF approaches that of the FMF1. This figure also validates the use of the Q-method EKF update, as the filters can accurately estimate both position and velocity (provided their respective optimal grid sizes). The velocity RMSE obtained with the EKF is also presented for comparison. Although the PMF variants achieve better accuracy than the EKF in estimating velocity, the difference is less pronounced than it is for position since velocity is estimated

linearly in both the PMFs and the EKF, resulting in similar performance.

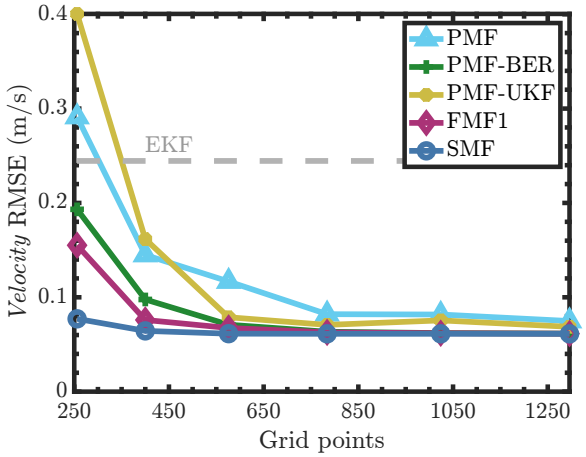

Fig. 12. Velocity RMSE for the different PMF variants as a function of grid points. The gray line represents the results obtained with an EKF and serves as a reference.

Figure 13 shows the SNEES for the different PMFs as a function of the total points of the grid. As in the Ikeda map, all SNEES values higher than 1×10^4 were treated as numerical instability and, therefore, disregarded. From this figure, it can be seen that the SMF is a more consistent filter, particularly at smaller grid sizes, which explains its high accuracy in such cases. The SNEES for this filter remains close to one for all grid sizes tested, reaching a SNEES of one at approximately 600 grid points. The PMF-UKF, PMF-BER, and FMF1 exhibit a similar behavior, with the FMF1 being more consistent throughout all grid sizes tested. The PMF struggles with the tested grid sizes, requiring more grid points to achieve a SNEES of one. Table III shows the final ($N = 1296$) RMSE and SNEES values for all filters used.

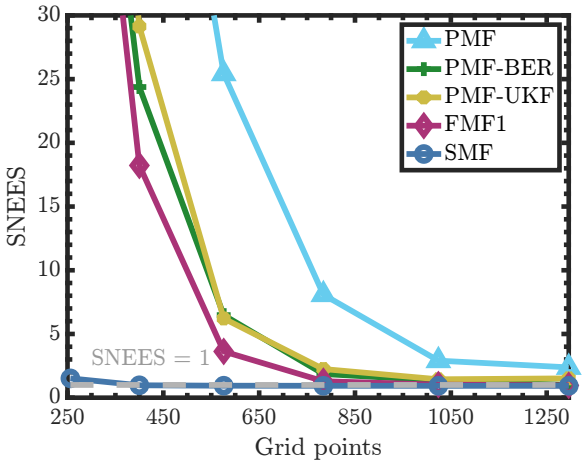

Fig. 13. SNEES for the different PMF variants as a function of grid points. The gray dashed line represents a SNEES value of one.

For this example, the results of the SMF are also compared to the PMF-DWC. In this case, only the analytic variant of the PMF-DWC is used since the dynamics for the position are linear and invertible. The number of grid points is set to $N = 256$, the lowest number tested in this section, and the grids are scaled to 5σ of their respective covariance matrices. In the case of the SMF, $\alpha = 0.3$. Just as in the previous section, the convolution in the SMF is performed without vectorization to ensure comparable runtime between the two filters. Table IV shows the RMSE and SNEES for the PMF-DWC$_a$ and the SMF obtained by simulating 1,000 trajectories.

TABLE III
Position and velocity RMSE and SNEES obtained for each filter ($N = 1296$)

Filter	Pos. RMSE (m)	Vel. RMSE (m/s)	SNEES
PMF	1.0542	0.0749	2.3776
PMF-BER	0.3690	0.0615	1.0027
PMF-UKF	0.5433	0.0687	1.5155
FMF1	0.3685	0.0615	0.9665
SMF	0.3637	0.0613	0.9301

TABLE IV
RMSE, SNEES, and runtime obtained for the SMF and the PMF-DWC ($N = 256$)

Filter	Pos. RMSE (m)	Vel. RMSE (m/s)	SNEES
PMF-DWC$_a$	3.6890	0.1318	26.4650
SMF	0.3731	0.0620	0.9097

From Table IV, it can be seen that in this example, performing the DWC also improves the accuracy and consistency of the standard PMF. However, the SMF results in a more accurate and consistent filter, as it handles low process noise while also improving grid placement. For this example, a speed-up is also observed, resulting in a runtime of 2.9348 seconds for the PMF-DWC$_a$ and 0.0212 seconds for the SMF, suggesting a difference of around two orders of magnitude, consistent with the results obtained for the Ikeda map.

X. CONCLUSIONS

This work introduces a new variant of the point mass filter, called the fusion mass filter. In this variant, a Gaussian sum filter update is performed before creating the predictive grid. The updated mean and covariance estimates are used to generate the grid, improving grid placement. An extension of the fusion mass filter, called the Silverman mass filter, is also presented. This extension uses kernel density estimation to more accurately approximate the predictive distribution, enabling a more effective Gaussian sum filter update and improving filter performance.

Through a bivariate example, it was shown that performing the Gaussian sum filter update prior to grid construction can result in a tighter, better placed grid, providing an accurate posterior distribution representation.

For this example, the Silverman mass filter was found to capture the tails of the posterior distribution better than the fusion mass filter, due to its conservative approach, achieved by using a scaled version of Silverman's rule of thumb.

Furthermore, the new variant was compared to the standard point mass filter and other grid design algorithms in two sequential filtering examples. The first example, which involved the Ikeda map, demonstrated improvements in accuracy and consistency compared to the other algorithms. The second example focused on terrain-relative navigation. For this example, the point mass filters were used to estimate the position of a vehicle, while a Q-Method EKF update was used to estimate velocity, reducing the computational cost of estimating both position and velocity nonlinearly. The results showed improved accuracy and consistency for the fusion mass filter and its extension compared to the other point mass filter variants.

Finally, future work could explore the performance of these algorithms with more efficient grid choices, such as sparse or non-uniform grids.

ACKNOWLEDGMENT

The authors would like to acknowledge the Ingenuity Mars helicopter for its inspiring, now completed, mission. This material is based on research sponsored by Air Force Office of Scientific Research (AFOSR) under award number: FA9550-22-1-0419.

REFERENCES

[1] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc, 2001.

[2] R. Bucy and K. Senne., "Digital synthesis of non-linear filters," *Automatica*, vol. 7, no. 3, pp. 287–298, 1971.

[3] M. Šimandl, J. Královec, and T. Söderström, "Anticipative grid design in point-mass approach to nonlinear state estimation," *IEEE Transactions on Automatic Control*, vol. 47, no. 4, pp. 699–702, 2002.

[4] ——, "Advanced point-mass method for nonlinear state estimation," *Automatica*, vol. 42, no. 7, pp. 1133–1145, 2006.

[5] N. Bergman, "A bayesian approach to terrain-aided navigation," *IFAC Proceedings Volumes*, vol. 30, no. 11, pp. 1457–1462, 1997.

[6] ——, "Recursive bayesian estimation: Navigation and tracking applications," Ph.D. dissertation, Linköping University, 1999.

[7] J. Duník, O. Straka, and J. Matoušek, "Conditional density driven grid design in point-mass filter," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 9180–9184.

[8] Y. Choe and C. G. Park, "Point-mass filtering with boundary flow and its application to terrain referenced navigation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 6, pp. 3600–3613, 2021.

[9] F. Giraldo-Grueso, A. A. Popov, and R. Zanetti, "Gaussian mixture-based point mass filtering," in *Proceedings of the 2024 27th International Conference on Information Fusion (FUSION)*, Venice, Italy, Jul. 2024.

[10] 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, pp. 4179–4197, 2022.

[11] A. A. Popov and R. Zanetti, "An adaptive covariance parameterization technique for the ensemble gaussian mixture filter," *arXiv preprint arXiv:2212.10323*, 2022.

[12] ——, "Ensemble gaussian mixture filtering with particle-localized covariances," in *Proceedings of the 2023 26th International Conference on Information Fusion (FUSION)*, Charleston, South Carolina, Jul. 2023.

[13] ——, "Ensemble-localized kernel density estimation with applications to the ensemble gaussian mixture filter," *arXiv preprint arXiv:2308.14143*, vol. 69, 2023.

[14] T. Ainscough, R. Zanetti, J. Christian, and P. D. Spanos, "Q-method extended kalman filter," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 4, pp. 752–760, 2015.

[15] J. Duník, O. Straka, J. Matoušek, and M. Brandner, "Accurate density-weighted convolution for point-mass filter and predictor," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 6, pp. 3574–3584, 2021.

[16] H. Sorenson and D. Alspach, "Recursive bayesian estimation using gaussian sums," *Automatica*, vol. 7, no. 4, pp. 465–479, 1971.

[17] D. Alspach and H. Sorenson, "Nonlinear bayesian estimation using gaussian sum approximations," *IEEE Transactions on Automatic Control*, vol. 17, no. 4, pp. 439–448, 1972.

[18] C.-K. Sung and S. J. Lee, "Reliable time propagation algorithms for pmf and rbpmf," *Sensors*, vol. 21, no. 6, p. 261, 2021.

[19] B. W. Silverman, *Density estimation for statistics and data analysis*. Routledge, 2018.

[20] E. Jaynes, "On the rationale of maximum-entropy methods," *Proceedings of the IEEE*, vol. 70, no. 9, pp. 939–952, 1982.

[21] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley-Interscience, 2006.

[22] 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, pp. 2741–2758, 1999.

[23] N. V. P. Janssen, J. S. Marron and W. Sarle, "Scale measures for bandwidth selection," *Journal of Nonparametric Statistics*, vol. 5, no. 4, pp. 359–380, 1995.

[24] K. Michaelson, A. A. Popov, and R. Zanetti, "Ensemble kalman filter with bayesian recursive update," in *Proceedings of the 2023 26th International Conference on Information Fusion (FUSION)*, Charleston, South Carolina, Jul. 2023.

[25] F. Daum, J. Huang, and A. Noushin, "Exact particle flow for nonlinear filters," in *Signal Processing, Sensor Fusion, and Target Recognition XIX*, vol. 7697, 2010, pp. 769–704.

[26] J. Diener, "2d minimal bounding box," https://www.mathworks.com/matlabcentral/fileexchange/31126-2d-minimal-bounding-box, 2024, retrieved December 7, 2024.

[27] J. Matoušek, J. Duník, and O. Straka, "Lagrangian grid-based filters with application to terrain-aided navigation," *IEEE Signal Processing Magazine. In press*, 2024.

[28] K. Ikeda, "Multiple-valued stationary state and its instability of the transmitted light by a ring cavity system," *Optics Communications*, vol. 30, no. 2, pp. 257–261, 1979.

[29] K. Ikeda, H. Daido, and O. Akimoto, "Optical turbulence: Chaotic behavior of transmitted light from a ring cavity," *Phys. Rev. Lett.*, vol. 45, pp. 709–712, 1980.

[30] D. F. Crouse and C. Lewis, "Consideration of particle flow filter implementations and biases," *Naval Research Laboratory Memo*, pp. 1–17, 2019.

[31] A. A. Popov and R. Zanetti, "An adaptive covariance parameterization technique for the ensemble gaussian mixture filter," *SIAM Journal on Scientific Computing*, vol. 46, no. 3, pp. A1949–A1971, 2024.

[32] J. Duník, M. Soták, M. Veselý, O. Straka, and W. Hawkinson, "Design of rao–blackwellized point-mass filter with application

in terrain aided navigation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 1, pp. 251–272, 2019.
[33] V. Šmídl and M. Gašperin, "Rao-blackwellized point mass filter for reliable state estimation," in *Proceedings of the 16th International Conference on Information Fusion*, 2013, pp. 312–318.
[34] X. Rong Li and V. Jilkov, "Survey of maneuvering target tracking. part i. dynamic models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, 2003.
[35] J. Matoušek, J. Duník, and M. Brandner, "Design of efficient point-mass filter with terrain aided navigation illustration," in *2023 26th International Conference on Information Fusion (FUSION)*, 2023.
[36] M. C. Malin, J. F. Bell III, B. A. Cantor, M. A. Caplinger, W. M. Calvin, R. T. Clancy, K. S. Edgett, L. Edwards, R. M. Haberle, P. B. James, S. W. Lee, M. A. Ravine, P. C. Thomas, and M. J. Wolff, "Context camera investigation on board the mars reconnaissance orbiter," *Journal of Geophysical Research: Planets*, vol. 112, no. E5, 2007.

Felipe Giraldo-Grueso obtained a B.S. in Mechanical Engineering and Physics from Universidad de Los Andes, Colombia, in 2021 and an M.S. in Aerospace Engineering from the University of Texas at Austin in 2023. He is currently a Ph.D. student in Dr. Zanetti's Nonlinear Estimation and Autonomy Research group. Felipe's research interests include navigation and guidance algorithms for space exploration.

Andrey A. Popov received his B.S. in Mathematics from RPI, Troy, NY, and his Ph.D. in Computer Science from Virginia Tech, Blacksburg, VA. He is currently an Assistant Professor at the University of Hawai'i at Mānoa. His research interests include state estimation/data assimilation, theory-guided machine learning, and data-driven methods for modeling and quantifying uncertainty in dynamical systems.

Renato Zanetti (Senior Member, IEEE) is an Associate Professor of Aerospace Engineering at The University of Texas at Austin. Prior to joining UT, he worked as an engineer at the NASA Johnson Space Center and at Draper Laboratory. Renato's research interests include nonlinear estimation, onboard navigation, and autonomous aerospace vehicles.