(Preprint) AAS 26-784

AN EFFICIENT FORMULATION OF GAUSSIAN PARTICLE FLOW FILTERS

Andrea Rigato[*], Renato Zanetti[†]

State estimation for strongly nonlinear dynamical systems remains a significant challenge in aerospace applications. Although classical particle filters asymptotically converge to exact Bayesian inference, they often suffer from weight degeneracy. Particle flow filters offer an alternative by transporting prior samples toward the posterior via the integration of a differential equation at each measurement update. This paper presents an efficient implementation of Gaussian particle flow filters that exploits a common structural feature of dynamical systems: only a subset of the state directly appears in the measurement function. By leveraging this structure, the proposed method substantially reduces the computational cost of pseudo-time flow integration while preserving estimation accuracy. We establish theoretical equivalence between the proposed implementation and several widely used Gaussian particle flow filters. Simulation results confirm that the method delivers accurate state estimates at substantially lower run time than standard implementations.

INTRODUCTION

Statistical estimation addresses the problem of inferring the state of a dynamical system from noisy measurements. When the system dynamics and measurement model are linear and the noise terms are Gaussian, the Kalman filter[1] provides the optimal solution in both the minimum mean square error (MMSE) and maximum a posteriori maximum a posteriori (MAP) senses.

For mildly nonlinear systems, several extensions of the Kalman filter, such as the extended Kalman filter (EKF)[2] and the unscented Kalman filter (UKF),[3] have been developed and are often effective in practice. However, these methods may diverge in the presence of strong nonlinearities in the dynamics or measurement models.

In the presence of strong nonlinearities, particle filters provide an approach that asymptotically converges to the true filtered distribution as the number of particles tends to infinity.[4,5] These methods approximate the posterior distribution with a set of weighted particles and can, in principle, handle arbitrary nonlinearities and non-Gaussian distributions. Nevertheless, they frequently suffer from weight degeneracy: after only a few iterations, a small number of particles retain significant weight while most others become negligible. Numerous techniques have been proposed to mitigate this issue, including the auxiliary particle filter (APF),[6] the extended Kalman particle filter (EKPF),[7] and the unscented particle filter (UPF).[7] Most of these approaches, however, still rely on a resampling step performed on the weighted posterior approximation.

[*]Ph.D. Student, Department of Aerospace Engineering and Engineering Mechanics, 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.

Particle flow filters provide an alternative solution to these limitations.[8] Instead of assigning and normalizing importance weights, these methods continuously transport samples from the prior distribution toward the posterior distribution through a pseudo-time parameter. At the end of the flow, all samples have equal weight and approximate draws from the posterior, avoiding both weight degeneracy and the need for resampling. Particle flow filters have shown particularly strong performance in high-dimensional and highly nonlinear problems, where conventional particle filters would require an impractically large number of particles to achieve comparable results.

The main drawback of particle flow filters is their computational cost: at each time step, an ordinary differential equation (ODE) or stochastic differential equation (SDE) must be integrated for every sample/particle, often with many integration steps.

In many aerospace applications only a subset of the state vector enters the measurement function. For example, estimating position and velocity of a vehicle is possible with measurements derived from position only, such as range and bearing angles. This structural property is ubiquitous in nonlinear filtering and has been exploited by several earlier approaches outside the particle flow framework. For instance, the q-Method was developed as a partial-update technique for the EKF,[9] and later extended to point mass filters.[10] Recently, a Rao-Blackwellized particle flow approach was proposed in which the states that enter the measurement equation are marginalized over the remaining dimensions.[11]

This work addresses the same scenario as reference [11] but from a different perspective that avoids marginalization and provides an efficient implementation of Gaussian particle flow filters that substantially reduces computational cost. Additionally, we establish theoretical equivalence between the proposed method and several widely used Gaussian particle flow filters.

The remainder of the paper is organized as follows. First, we review the foundational theory of particle flow filters. Next, we introduce the proposed efficient Gaussian particle flow implementation and derive it in detail. We then establish theoretical equivalence with three representative Gaussian particle flow methods. Finally, we present numerical results to evaluate performance and accuracy and conclude the paper.

PARTICLE FLOW FILTERS

Consider the following discrete-time nonlinear system,

$$\begin{aligned} \boldsymbol{x}_k &= \boldsymbol{f}_{k-1}(\boldsymbol{x}_{k-1}) + \boldsymbol{v}_{k-1}, \\ \boldsymbol{y}_k &= \boldsymbol{h}_k(\boldsymbol{x}_k) + \boldsymbol{\eta}_k, \end{aligned} \tag{1}$$

where $\boldsymbol{x}_k$ is the state at time k, $\boldsymbol{y}_k$ denotes the measurements available at time k, $\boldsymbol{f}_{k-1}(\cdot)$ is the state transition function, $\boldsymbol{h}_k(\cdot)$ is the measurement function, and $\boldsymbol{v}_{k-1}$ and $\boldsymbol{\eta}_k$ are the process and measurement noises, respectively.

The state estimation problem is described by the Bayesian recursive relations (BRR). Starting from an initial posterior density $p(\boldsymbol{x}_{k-1} \mid \boldsymbol{y}_{k-1})$, the prior density at the next time step is obtained via the Chapman–Kolmogorov equation:[2]

$$p(\boldsymbol{x}_k \mid \boldsymbol{y}_{k-1}) = \int_{\mathcal{D}} p(\boldsymbol{x}_k \mid \boldsymbol{x}_{k-1})\, p(\boldsymbol{x}_{k-1} \mid \boldsymbol{y}_{k-1})\, d\boldsymbol{x}_{k-1}, \tag{2}$$

where $\mathcal{D}$ is the domain of the state space $\boldsymbol{x}_{k-1}$. This prior probability density function (pdf) is then updated using Bayes' rule:[2]

$$p(\boldsymbol{x}_k \mid \boldsymbol{y}_k) = \frac{p(\boldsymbol{y}_k \mid \boldsymbol{x}_k, \boldsymbol{y}_{k-1})\, p(\boldsymbol{x}_k \mid \boldsymbol{y}_{k-1})}{p(\boldsymbol{y}_k \mid \boldsymbol{y}_{k-1})}. \tag{3}$$

Although analytical solutions exist in the linear case and under the assumption of Gaussian or Gaussian mixture densities,[1,12] the problem becomes intractable in the nonlinear case.

Particle flow filters address the BRR by approximating both the prior and posterior densities using collections of samples. In this work, such a collection of samples is referred to as an *ensemble*. Although the term *particle* appears in the name of the filter, the authors deliberately avoid using *particle(s)* to describe these samples. This choice helps clearly distinguish the present approach from conventional particle filters,[4] in which particles are associated with importance weights.

Consider N samples $\boldsymbol{x}^{(i)}_{k-1|k-1}$ obtained from the posterior distribution $p(\boldsymbol{x}_{k-1} \mid \boldsymbol{y}_{k-1})$. The ensemble at time $k-1$ is defined as:

$$\mathbf{X}_{k-1|k-1} = \left[\boldsymbol{x}^{(1)}_{k-1|k-1}, \boldsymbol{x}^{(2)}_{k-1|k-1}, \ldots, \boldsymbol{x}^{(N)}_{k-1|k-1}\right]. \tag{4}$$

In the limit $N \to \infty$, this finite ensemble is assumed to exactly represent the posterior distribution $p(\boldsymbol{x}_{k-1} \mid \boldsymbol{y}_{k-1})$. The samples are then propagated forward through the system dynamics, each driven by an independent realization of the process noise:

$$\boldsymbol{x}^{(i)}_{k|k-1} = \boldsymbol{f}_{k-1}\left(\boldsymbol{x}^{(i)}_{k-1|k-1}\right) + \boldsymbol{v}^{(i)}_{k-1}, \quad i = 1, \ldots, N. \tag{5}$$

The resulting predicted ensemble

$$\mathbf{X}_{k|k-1} = \left[\boldsymbol{x}^{(1)}_{k|k-1}, \boldsymbol{x}^{(2)}_{k|k-1}, \ldots, \boldsymbol{x}^{(N)}_{k|k-1}\right], \tag{6}$$

therefore constitutes, in the large-sample limit, an exact representation of the prior distribution $p(\boldsymbol{x}_k \mid \boldsymbol{y}_{k-1})$. This prediction step is shared with other sample-based methods, such as the bootstrap particle filter[4] and the ensemble Kalman filter (EnKF).[13] The distinguishing feature of particle flow filters lies in the update step. Rather than performing a discrete correction, particle flow filters transport the samples continuously from the prior $\mathbf{X}_{k|k-1}$ to the posterior $\mathbf{X}_{k|k}$ along a smooth path.

While some particle flow filter variants rely on techniques such as Stein variational gradient descent,[14] projected cumulative distributions,[15] or minimization of the KL divergence,[16] the majority of particle flow filters are formulated using the concept of log-homotopy.[17]

To introduce the log-homotopy idea, the following simplified notation is adopted:

$$\begin{aligned} p(\boldsymbol{x}_k \mid \boldsymbol{y}_k) &= p(\boldsymbol{x}), \qquad p(\boldsymbol{y}_k \mid \boldsymbol{x}_k, \boldsymbol{y}_{k-1}) = l(\boldsymbol{x}) \\ p(\boldsymbol{x}_k \mid \boldsymbol{y}_{k-1}) &= g(\boldsymbol{x}), \qquad p(\boldsymbol{y}_k \mid \boldsymbol{y}_{k-1}) = K. \end{aligned} \tag{7}$$

Consider the scalar homotopy parameter $\lambda \in [0, 1]$; it is then possible to define the log-homotopy distribution as:

$$\log p(\boldsymbol{x}, \lambda) = \log g(\boldsymbol{x}) + \lambda \log l(\boldsymbol{x}) - \log K(\lambda), \tag{8}$$

where $K(\lambda)$ is a normalization function such that $\exp(\log p(\boldsymbol{x}, \lambda))$ is a valid pdf for all λ. Clearly, $p(\boldsymbol{x}, \lambda = 0)$ is the prior distribution, and $p(\boldsymbol{x}, \lambda = 1)$ is the posterior distribution. The core idea is to evolve the ensemble from the prior $\mathbf{X}_{k|k-1}$ to the posterior $\mathbf{X}_{k|k}$ by following a stochastic flow governed by:

$$d\boldsymbol{x} = \mathbf{d}(\boldsymbol{x}, \lambda)\, d\lambda + \mathbf{G}(\boldsymbol{x}, \lambda) d\boldsymbol{w}, \tag{9}$$

where $\mathbf{d}(\cdot)$ is the drift function, $\mathbf{G}(\cdot)$ is the diffusion matrix function, and $d\boldsymbol{w}$ is a Wiener process. The corresponding probability density evolves according to the Fokker–Planck equation:[18]

$$\frac{\partial\, p(\boldsymbol{x}, \lambda)}{\partial \lambda} = -\mathrm{div}(p(\boldsymbol{x}, \lambda)\, \mathbf{d}(\boldsymbol{x}, \lambda)) + \frac{1}{2} \nabla_{\boldsymbol{x}}^T \left(p(\boldsymbol{x}, \lambda)\, \mathbf{Q}(\boldsymbol{x}, \lambda)\right) \nabla_{\boldsymbol{x}}, \tag{10}$$

where $\boldsymbol{Q}(\boldsymbol{x}, \lambda) = \mathbf{G}(\boldsymbol{x}, \lambda)\mathbf{G}(\boldsymbol{x}, \lambda)^T$, $\mathrm{div}(\cdot)$ is the divergence operator, and $\nabla_{\boldsymbol{x}}$ denotes the gradient with respect to $\boldsymbol{x}$.

By combining the log-homotopy construction eq. (8) with the Fokker–Planck dynamics eq. (10), one obtains a partial differential equation that can be solved for the drift function $\mathbf{d}(\cdot)$ and the diffusion function $\mathbf{G}(\cdot)$ such that the resulting flow maps the prior density at $\lambda = 0$ to the posterior density at $\lambda = 1$.

In practice, deriving closed-form expressions for $\mathbf{d}(\cdot)$ and $\mathbf{G}(\cdot)$ requires assumptions on the functional form of the prior density and the likelihood. In this work, we consider Gaussian particle flow filters, in which the prior density $p(\boldsymbol{x}_k \mid \boldsymbol{y}_{k-1})$ is assumed to be Gaussian.

Once the SDE in eq. (9) is specified under these assumptions, the posterior ensemble $\mathbf{X}_{k|k}$ is obtained by integrating the flow from $\lambda = 0$ to $\lambda = 1$:

$$\mathbf{X}_{k|k} = \mathbf{X}_{k|k-1} + \int_0^1 \mathbf{d}\big(\mathbf{X}(\lambda), \lambda\big)\, d\lambda + \int_0^1 \mathbf{G}\big(\mathbf{X}(\lambda), \lambda\big)\, d\mathbf{w}(\lambda). \tag{11}$$

EFFICIENT EVALUATION APPROACH FOR GAUSSIAN PARTICLE FLOW FILTERS

Particle flow filters have proven highly effective, particularly in high-dimensional systems where the weight degeneracy problem severely degrades the performance of conventional particle filters.[19] However, particle flow filters are computationally expensive: at each time step, a stochastic differential equation (or an ordinary differential equation in the zero-diffusion case where $\mathbf{G}(\cdot) = 0$) must be numerically integrated along the pseudo-time parameter $\lambda \in [0, 1]$.

Moreover, in many practical applications, like weather forecasting, geophysical modeling, and aerospace systems, only a subset of the state vector is directly and instantaneously observable through the measurement function. This observation naturally raises the central question addressed in this work:

In high-dimensional systems with a large number of instantaneously unobservable states, can the flow be computed more efficiently?

As explained in the remainder of the paper, the answer is affirmative.

To simplify notation, since the flow occurs in pseudo-time at a fixed time index k, we drop the explicit dependence on k and emphasize the dependence on the pseudo-time parameter λ. Accordingly, the following notation is adopted:

$$\mathbf{X}_{k|k-1} = \mathbf{X}(0) = \left[\boldsymbol{x}^{(1)}(0),\, \boldsymbol{x}^{(2)}(0),\, \ldots,\, \boldsymbol{x}^{(N)}(0)\right], \tag{12}$$

$$\mathbf{X}_{k|k} = \mathbf{X}(1) = \left[\boldsymbol{x}^{(1)}(1),\, \boldsymbol{x}^{(2)}(1),\, \ldots,\, \boldsymbol{x}^{(N)}(1)\right]. \tag{13}$$

Furthermore, the state vector is partitioned into two parts: one that enters the measurement function and one that does not. These are referred to as *measured* and *unmeasured* states, respectively. Accordingly, the state is expressed as:

$$\boldsymbol{x}(\lambda) = \begin{bmatrix} \boldsymbol{x}^{\mathrm{m}}(\lambda) \\ \boldsymbol{x}^{\mathrm{u}}(\lambda) \end{bmatrix}, \qquad \boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}^{\mathrm{m}}) + \boldsymbol{\eta}. \tag{14}$$

Given this partitioning, the integrated flow equations split into two coupled equations for the measured and unmeasured parts:

$$\mathbf{X}^{\mathrm{m}}(1) = \mathbf{X}^{\mathrm{m}}(0) + \int_0^1 \mathbf{d}^{\mathrm{m}}(\mathbf{X}(\lambda), \lambda)\ d\lambda + \int_0^1 \mathbf{G}^{\mathrm{m}}(\mathbf{X}(\lambda), \lambda)\ d\mathbf{w}(\lambda), \tag{15}$$

$$\mathbf{X}^{\mathrm{u}}(1) = \mathbf{X}^{\mathrm{u}}(0) + \int_0^1 \mathbf{d}^{\mathrm{u}}(\mathbf{X}(\lambda), \lambda)\ d\lambda + \int_0^1 \mathbf{G}^{\mathrm{u}}(\mathbf{X}(\lambda), \lambda)\ d\mathbf{w}(\lambda), \tag{16}$$

with the full ensemble, drift and diffusion vectors partitioned conformably

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}^{\mathrm{m}} \\ \mathbf{X}^{\mathrm{u}} \end{bmatrix}, \qquad \mathbf{d}(\cdot) = \begin{bmatrix} \mathbf{d}^{\mathrm{m}}(\cdot) \\ \mathbf{d}^{\mathrm{u}}(\cdot) \end{bmatrix}, \qquad \mathbf{G}(\cdot) = \begin{bmatrix} \mathbf{G}^{\mathrm{m}}(\cdot) \\ \mathbf{G}^{\mathrm{u}}(\cdot) \end{bmatrix}. \tag{17}$$

This decomposition of the flow is closely related to the structure of the Bayesian update. Suppressing the explicit time index k and the conditioning on $\boldsymbol{y}_{k-1}$ in eq. (3), the posterior distribution can be expressed as:

$$p(\boldsymbol{x} \mid \boldsymbol{y}) = p(\boldsymbol{x}^{\mathrm{m}}, \boldsymbol{x}^{\mathrm{u}} \mid \boldsymbol{y}), \tag{18}$$

$$= \frac{p(\boldsymbol{y} \mid \boldsymbol{x}^{\mathrm{m}}, \boldsymbol{x}^{\mathrm{u}})\, p(\boldsymbol{x}^{\mathrm{m}}, \boldsymbol{x}^{\mathrm{u}})}{p(\boldsymbol{y})}, \tag{19}$$

$$= \frac{p(\boldsymbol{y} \mid \boldsymbol{x}^{\mathrm{m}})\, p(\boldsymbol{x}^{\mathrm{m}})\, p(\boldsymbol{x}^{\mathrm{u}} \mid \boldsymbol{x}^{\mathrm{m}})}{p(\boldsymbol{y})}, \tag{20}$$

$$= \frac{p(\boldsymbol{y} \mid \boldsymbol{x}^{\mathrm{m}})\, p(\boldsymbol{x}^{\mathrm{m}})}{p(\boldsymbol{y})}\, p(\boldsymbol{x}^{\mathrm{u}} \mid \boldsymbol{x}^{\mathrm{m}}), \tag{21}$$

$$= p(\boldsymbol{x}^{\mathrm{m}} \mid \boldsymbol{y})\, p(\boldsymbol{x}^{\mathrm{u}} \mid \boldsymbol{x}^{\mathrm{m}}). \tag{22}$$

This factorization reveals that, when only a subset of the state is measured, the measurement information directly affects only the update of the measured component $\boldsymbol{x}^{\mathrm{m}}$; moreover, this update does not depend on the unmeasured component $\boldsymbol{x}^{\mathrm{u}}$. The update of the unmeasured component $\boldsymbol{x}^{\mathrm{u}}$ occurs solely through its statistical dependence on the measured part $\boldsymbol{x}^{\mathrm{m}}$.

A direct consequence of this structure is that the flow of the measured ensemble $\mathbf{X}^{\mathrm{m}}$ in eq. (15) depends only on $\mathbf{X}^{\mathrm{m}}$ and not on the unmeasured ensemble $\mathbf{X}^{\mathrm{u}}$. The update equation for the measured part therefore simplifies to

$$\mathbf{X}^{\mathrm{m}}(1) = \mathbf{X}^{\mathrm{m}}(0) + \int_0^1 \mathbf{d}^{\mathrm{m}}(\mathbf{X}^{\mathrm{m}}(\lambda), \lambda)\ d\lambda + \int_0^1 \mathbf{G}^{\mathrm{m}}(\mathbf{X}^{\mathrm{m}}(\lambda), \lambda)\ d\mathbf{w}(\lambda). \tag{23}$$

In this work, we consider Gaussian particle flow filters. Assuming the prior distribution is jointly Gaussian,

$$p(\boldsymbol{x}^{\mathrm{m}}, \boldsymbol{x}^{\mathrm{u}}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{x}^{\mathrm{m}} \\ \boldsymbol{x}^{\mathrm{u}} \end{bmatrix}; \begin{bmatrix} \bar{\boldsymbol{x}}^{\mathrm{m}} \\ \bar{\boldsymbol{x}}^{\mathrm{u}} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}^{\mathrm{mm}} & \boldsymbol{\Sigma}^{\mathrm{mu}} \\ \boldsymbol{\Sigma}^{\mathrm{um}} & \boldsymbol{\Sigma}^{\mathrm{uu}} \end{bmatrix} \right), \tag{24}$$

it is possible to update the unmeasured ensemble using a conditional Gaussian update:

$$\mathbf{X}^{\mathrm{u}}(1) = \mathbf{X}^{\mathrm{u}}(0) + \boldsymbol{\Sigma}^{\mathrm{um}} \left(\boldsymbol{\Sigma}^{\mathrm{mm}}\right)^{-1} \left(\mathbf{X}^{\mathrm{m}}(1) - \mathbf{X}^{\mathrm{m}}(0)\right). \tag{25}$$

Algorithm 1 summarizes the prediction and measurement update steps of the efficient Gaussian particle flow filter.

Clearly, if eqs. (16), (23) and (25) hold simultaneously, then the drift and diffusion terms for the measured and unmeasured parts must satisfy the following conditions:

$$\mathbf{d}^{\mathrm{u}}(\mathbf{X}(\lambda), \lambda) = \mathbf{d}^{\mathrm{u}}(\mathbf{X}^{\mathrm{m}}(\lambda), \lambda) = \boldsymbol{\Sigma}^{\mathrm{um}} \left(\boldsymbol{\Sigma}^{\mathrm{mm}}\right)^{-1} \mathbf{d}^{\mathrm{m}}(\mathbf{X}^{\mathrm{m}}(\lambda), \lambda), \tag{26}$$

$$\mathbf{G}^{\mathrm{u}}(\mathbf{X}(\lambda), \lambda) = \mathbf{G}^{\mathrm{u}}(\mathbf{X}^{\mathrm{m}}(\lambda), \lambda) = \boldsymbol{\Sigma}^{\mathrm{um}} \left(\boldsymbol{\Sigma}^{\mathrm{mm}}\right)^{-1} \mathbf{G}^{\mathrm{m}}(\mathbf{X}^{\mathrm{m}}(\lambda), \lambda). \tag{27}$$

Algorithm 1: Efficient Gaussian Particle Flow Filter

Input: $p(\boldsymbol{x}_0)$, N, $\boldsymbol{f}_{k-1}(\cdot)$, $\boldsymbol{h}_k(\cdot)$, $\boldsymbol{y}_k$, $\boldsymbol{Q}_{k-1}$, $\boldsymbol{R}_k$, $\mathbf{d}(\cdot)$, $\mathbf{G}(\cdot)$

Output: $\boldsymbol{x}_{k|k}$, $\boldsymbol{\Sigma}_{k|k}$

Initialization: Sample ensemble members $\boldsymbol{x}_{0|0}^{(i)} \sim p(\boldsymbol{x}_0)$ for $i = 1, \ldots, N$

for $k = 1, 2, \ldots$ **do**

- **Time update:**
- **for** $i = 1$ **to** N **do**
  - $\boldsymbol{v}_{k-1}^{(i)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}_{k-1})$,
  - $\boldsymbol{x}_{k|k-1}^{(i)} \leftarrow \boldsymbol{f}_{k-1}(\boldsymbol{x}_{k-1|k-1}^{(i)}) + \boldsymbol{v}_{k-1}^{(i)}$.
- **Compute predicted moments:**
- $\boldsymbol{x}_{k|k-1} \leftarrow \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_{k|k-1}^{(i)}$,
- $\boldsymbol{\Sigma}_{k|k-1} \leftarrow \frac{1}{N} \sum_{i=1}^{N} \left(\boldsymbol{x}_{k|k-1}^{(i)} - \boldsymbol{x}_{k|k-1}\right) \left(\boldsymbol{x}_{k|k-1}^{(i)} - \boldsymbol{x}_{k|k-1}\right)$,
- $\boldsymbol{\Sigma}_{k|k-1} = \begin{bmatrix} \boldsymbol{\Sigma}^{\mathrm{mm}} & \boldsymbol{\Sigma}^{\mathrm{mu}} \\ \boldsymbol{\Sigma}^{\mathrm{um}} & \boldsymbol{\Sigma}^{\mathrm{uu}} \end{bmatrix}$,
- **Measurement update:**
- **for** $i = 1$ **to** N **do**
  - $\boldsymbol{x}_{k|k}^{\mathrm{m},(i)} \leftarrow \boldsymbol{x}_{k|k-1}^{\mathrm{m},(i)} + \int_0^1 \mathbf{d}^{\mathrm{m}}\left(\boldsymbol{x}^{\mathrm{m},(i)}, \lambda\right) \, d\lambda + \int_0^1 \mathbf{G}^{\mathrm{m}}\left(\boldsymbol{x}^{\mathrm{m},(i)}, \lambda\right) \, d\mathbf{w}(\lambda)$,
  - $\boldsymbol{x}_{k|k}^{\mathrm{u},(i)} \leftarrow \boldsymbol{x}_{k|k-1}^{\mathrm{u},(i)} + \boldsymbol{\Sigma}^{\mathrm{um}} \left(\boldsymbol{\Sigma}^{\mathrm{mm}}\right)^{-1} \left(\boldsymbol{x}_{k|k}^{\mathrm{m},(i)} - \boldsymbol{x}_{k|k-1}^{\mathrm{m},(i)}\right)$.
- **Compute filtered moments:**
- $\boldsymbol{x}_{k|k} \leftarrow \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_{k|k}^{(i)}$,
- $\boldsymbol{\Sigma}_{k|k} \leftarrow \frac{1}{N} \sum_{i=1}^{N} \left(\boldsymbol{x}_{k|k}^{(i)} - \boldsymbol{x}_{k|k}\right) \left(\boldsymbol{x}_{k|k}^{(i)} - \boldsymbol{x}_{k|k}\right)$.

THEORETICAL EQUIVALENCE RESULTS FOR SELECTED GAUSSIAN PARTICLE FLOW FILTERS

In this section, we show that the consistency conditions derived in eqs. (26) and (27) hold for three Gaussian particle flow filters: the exact flow filter,[19] the Gromov flow filter,[20] and the burnished flow filter.[21] As explained previously, when these relationships are satisfied, flowing the full ensemble using eq. (11) becomes mathematically equivalent to flowing only the measured ensemble using eq. (23) and updating the unmeasured ensemble via the conditional Gaussian update in eq. (25).

Although the results extend to the nonlinear case without loss of generality, the proofs presented below focus on the linear version of the respective filters, where the measurement model is assumed linear with Gaussian noise:

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{\eta} = \begin{bmatrix}\boldsymbol{H}^{\mathrm{m}} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}^{\mathrm{m}}\\ \boldsymbol{x}^{\mathrm{u}}\end{bmatrix} + \boldsymbol{\eta}, \quad \boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R}). \tag{28}$$

The same conclusions hold in the nonlinear setting, as the key structural properties are preserved.

Exact flow filter

The exact flow filter is a diffusion-free particle flow filter (i.e., $\mathbf{G}(\cdot) = 0$) in which the drift function is given by:[19]

$$\mathbf{d}(\mathbf{X}(\lambda), \lambda) = \boldsymbol{A}(\lambda)\,\mathbf{X}(\lambda) + \boldsymbol{b}(\lambda), \tag{29}$$

where

$$\boldsymbol{A}(\lambda) = -\frac{1}{2}\boldsymbol{\Sigma}\boldsymbol{H}^T(\lambda\,\boldsymbol{H}\boldsymbol{\Sigma}\boldsymbol{H}^T + \boldsymbol{R})^{-1}\boldsymbol{H}, \tag{30}$$

$$\boldsymbol{b}(\lambda) = (\boldsymbol{I} + 2\lambda\,\boldsymbol{A})\left[(\boldsymbol{I} + \lambda\,\boldsymbol{A})\boldsymbol{\Sigma}\boldsymbol{H}^T\boldsymbol{R}^{-1}\boldsymbol{y} + \boldsymbol{A}\bar{\boldsymbol{x}}\right]. \tag{31}$$

Now consider the partitioning of the prior covariance matrix and the structure of the measurement function:

$$\boldsymbol{\Sigma} = \begin{bmatrix}\boldsymbol{\Sigma}^{\mathrm{mm}} & \boldsymbol{\Sigma}^{\mathrm{mu}}\\ \boldsymbol{\Sigma}^{\mathrm{um}} & \boldsymbol{\Sigma}^{\mathrm{uu}}\end{bmatrix}, \qquad \boldsymbol{H} = \begin{bmatrix}\boldsymbol{H}^{\mathrm{m}} & \mathbf{0}\end{bmatrix}, \tag{32}$$

and denote the innovation covariance matrix at pseudo-time λ as

$$\lambda\,\boldsymbol{H}\boldsymbol{\Sigma}\boldsymbol{H}^T + \boldsymbol{R} = \lambda\boldsymbol{H}^{\mathrm{m}}\boldsymbol{\Sigma}^{\mathrm{mm}}(\boldsymbol{H}^{\mathrm{m}})^T + \boldsymbol{R} = \boldsymbol{S}_\lambda. \tag{33}$$

Substituting these partitioned forms yields

$$\boldsymbol{A}(\lambda) = -\frac{1}{2}\begin{bmatrix}\boldsymbol{\Sigma}^{\mathrm{mm}}(\boldsymbol{H}^{\mathrm{m}})^T\boldsymbol{S}_\lambda^{-1}\boldsymbol{H}^{\mathrm{m}} & \mathbf{0}\\ \boldsymbol{\Sigma}^{\mathrm{um}}(\boldsymbol{H}^{\mathrm{m}})^T\boldsymbol{S}_\lambda^{-1}\boldsymbol{H}^{\mathrm{m}} & \mathbf{0}\end{bmatrix}. \tag{34}$$

Defining $\boldsymbol{A}^{\mathrm{m}}(\lambda) = -\frac{1}{2}\boldsymbol{\Sigma}^{\mathrm{mm}}(\boldsymbol{H}^{\mathrm{m}})^T\boldsymbol{S}_\lambda^{-1}\boldsymbol{H}^{\mathrm{m}}$ and $\boldsymbol{K} = \boldsymbol{\Sigma}^{\mathrm{um}}\,(\boldsymbol{\Sigma}^{\mathrm{mm}})^{-1}$, the matrix $\boldsymbol{A}(\lambda)$ takes the block structure

$$\boldsymbol{A}(\lambda) = \begin{bmatrix}\boldsymbol{A}^{\mathrm{m}}(\lambda) & \mathbf{0}\\ \boldsymbol{K}\boldsymbol{A}^{\mathrm{m}}(\lambda) & \mathbf{0}\end{bmatrix}. \tag{35}$$

Turning now to the bias term $\boldsymbol{b}(\lambda)$, the partitioned expressions lead to

$$\begin{aligned}\boldsymbol{b}(\lambda) &= \begin{bmatrix}\boldsymbol{I} + 2\lambda\boldsymbol{A}^{\mathrm{m}} & \mathbf{0}\\ 2\lambda\boldsymbol{K}\boldsymbol{A}^{\mathrm{m}} & \boldsymbol{I}\end{bmatrix}\left[\begin{bmatrix}\boldsymbol{I} + \lambda\,\boldsymbol{A}^{\mathrm{m}} & \mathbf{0}\\ \lambda\boldsymbol{K}\boldsymbol{A}^{\mathrm{m}} & \boldsymbol{I}\end{bmatrix}\boldsymbol{\Sigma}\boldsymbol{H}^T\boldsymbol{R}^{-1}\boldsymbol{y} + \boldsymbol{A}\bar{\boldsymbol{x}}\right], &(36)\\ &= \begin{bmatrix}\boldsymbol{I} + 2\lambda\,\boldsymbol{A}^{\mathrm{m}} & \mathbf{0}\\ 2\lambda\boldsymbol{K}\boldsymbol{A}^{\mathrm{m}} & \boldsymbol{I}\end{bmatrix}\left[\begin{bmatrix}\boldsymbol{\Sigma}^{\mathrm{mm}}(\boldsymbol{H}^{\mathrm{m}})^T + \lambda\,\boldsymbol{A}^{\mathrm{m}}\boldsymbol{\Sigma}^{\mathrm{mm}}(\boldsymbol{H}^{\mathrm{m}})^T\\ \boldsymbol{\Sigma}^{\mathrm{um}}(\boldsymbol{H}^{\mathrm{m}})^T + \lambda\,\boldsymbol{K}\boldsymbol{A}^{\mathrm{m}}\boldsymbol{\Sigma}^{\mathrm{mm}}(\boldsymbol{H}^{\mathrm{m}})^T\end{bmatrix}\boldsymbol{R}^{-1}\boldsymbol{y} + \begin{bmatrix}\boldsymbol{A}^{\mathrm{m}}\bar{\boldsymbol{x}}^{\mathrm{m}}\\ \boldsymbol{K}\boldsymbol{A}^{\mathrm{m}}\bar{\boldsymbol{x}}^{\mathrm{m}}\end{bmatrix}\right]. &(37)\end{aligned}$$

Denote the common observable contribution as

$$\boldsymbol{C} = \left[\boldsymbol{\Sigma}^{\mathrm{mm}}(\boldsymbol{H}^{\mathrm{m}})^T + \lambda\,\boldsymbol{A}^{\mathrm{m}}\boldsymbol{\Sigma}^{\mathrm{mm}}(\boldsymbol{H}^{\mathrm{m}})^T\right]\boldsymbol{R}^{-1}\boldsymbol{y} + \boldsymbol{A}^{\mathrm{m}}\bar{\boldsymbol{x}}^{\mathrm{m}}. \tag{38}$$

This allows the bias term to be rewritten as

$$\boldsymbol{b}(\lambda) = \begin{bmatrix} \boldsymbol{I} + 2\lambda\, \boldsymbol{A}^{\mathrm{m}} & \boldsymbol{0} \\ 2\lambda \boldsymbol{K} \boldsymbol{A}^{\mathrm{m}} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{K}\boldsymbol{C} \end{bmatrix}, \tag{39}$$

$$= \begin{bmatrix} \boldsymbol{C} + 2\lambda\, \boldsymbol{A}^{\mathrm{m}} \boldsymbol{C} \\ \boldsymbol{K}\boldsymbol{C} + 2\lambda \boldsymbol{K} \boldsymbol{A}^{\mathrm{m}} \boldsymbol{C} \end{bmatrix}, \tag{40}$$

$$= \begin{bmatrix} \boldsymbol{b}^{\mathrm{m}}(\lambda) \\ \boldsymbol{K}\boldsymbol{b}^{\mathrm{m}}(\lambda) \end{bmatrix}. \tag{41}$$

Combining the results obtained for both $\boldsymbol{A}(\lambda)$ and $\boldsymbol{b}(\lambda)$, the drift function becomes

$$\mathbf{d}(\mathbf{X}(\lambda), \lambda) = \begin{bmatrix} \boldsymbol{A}^{\mathrm{m}}(\lambda)\, \mathbf{X}^{\mathrm{m}}(\lambda) + \boldsymbol{b}^{\mathrm{m}}(\lambda) \\ \boldsymbol{K} \boldsymbol{A}^{\mathrm{m}}(\lambda)\, \mathbf{X}^{\mathrm{m}}(\lambda) + \boldsymbol{K}\boldsymbol{b}^{\mathrm{m}}(\lambda) \end{bmatrix} = \begin{bmatrix} \mathbf{d}^{\mathrm{m}}(\mathbf{X}^{\mathrm{m}}(\lambda), \lambda) \\ \boldsymbol{\Sigma}^{\mathrm{um}} \left(\boldsymbol{\Sigma}^{\mathrm{mm}}\right)^{-1} \mathbf{d}^{\mathrm{m}}(\mathbf{X}^{\mathrm{m}}(\lambda), \lambda) \end{bmatrix}, \tag{42}$$

which confirms the validity of eq. (26).

Gromov flow filter

The Gromov flow filter is a stochastic particle flow filter, in which the drift and diffusion functions are given by:[20]

$$\mathbf{d}(\mathbf{X}(\lambda), \lambda) = -\left(\boldsymbol{\Sigma}^{-1} + \lambda\, \boldsymbol{H}^T \boldsymbol{R}^{-1} \boldsymbol{H}\right)^{-1} \boldsymbol{H}^T \boldsymbol{R}^{-1} \left(\boldsymbol{H}\, \mathbf{X}(\lambda) - \boldsymbol{y}\right), \tag{43}$$

$$\mathbf{G}(\mathbf{X}(\lambda), \lambda) = \left(\boldsymbol{\Sigma}^{-1} + \lambda\, \boldsymbol{H}^T \boldsymbol{R}^{-1} \boldsymbol{H}\right)^{-1} \boldsymbol{H}^T \boldsymbol{R}^{-1/2}. \tag{44}$$

Both expressions share the common term

$$\boldsymbol{L}(\lambda) = \left(\boldsymbol{\Sigma}^{-1} + \lambda\, \boldsymbol{H}^T \boldsymbol{R}^{-1} \boldsymbol{H}\right)^{-1} \boldsymbol{H}^T. \tag{45}$$

Applying the matrix inversion lemma[22] yields

$$\boldsymbol{L}(\lambda) = \boldsymbol{\Sigma} \boldsymbol{H}^T - \lambda\, \boldsymbol{\Sigma} \boldsymbol{H}^T \left(\lambda\, \boldsymbol{H} \boldsymbol{\Sigma} \boldsymbol{H}^T + \boldsymbol{R}\right)^{-1} \boldsymbol{H} \boldsymbol{\Sigma} \boldsymbol{H}^T. \tag{46}$$

Substituting the partitioned forms of the prior covariance matrix and measurement matrix from eq. (32), together with the definition of the innovation covariance $\boldsymbol{S}_\lambda$ from eq. (33), gives

$$\boldsymbol{L}(\lambda) = \boldsymbol{\Sigma} \boldsymbol{H}^T - \lambda\, \boldsymbol{\Sigma} \boldsymbol{H}^T \left(\lambda\, \boldsymbol{H} \boldsymbol{\Sigma} \boldsymbol{H}^T + \boldsymbol{R}\right)^{-1} \boldsymbol{H} \boldsymbol{\Sigma} \boldsymbol{H}^T, \tag{47}$$

$$= \begin{bmatrix} \boldsymbol{\Sigma}^{\mathrm{mm}} \left(\boldsymbol{H}^{\mathrm{m}}\right)^T \\ \boldsymbol{\Sigma}^{\mathrm{um}} \left(\boldsymbol{H}^{\mathrm{m}}\right)^T \end{bmatrix} - \lambda \begin{bmatrix} \boldsymbol{\Sigma}^{\mathrm{mm}} \left(\boldsymbol{H}^{\mathrm{m}}\right)^T \boldsymbol{S}_\lambda^{-1} \boldsymbol{H}^{\mathrm{m}} \boldsymbol{\Sigma}^{\mathrm{mm}} \left(\boldsymbol{H}^{\mathrm{m}}\right)^T \\ \boldsymbol{\Sigma}^{\mathrm{um}} \left(\boldsymbol{H}^{\mathrm{m}}\right)^T \boldsymbol{S}_\lambda^{-1} \boldsymbol{H}^{\mathrm{m}} \boldsymbol{\Sigma}^{\mathrm{mm}} \left(\boldsymbol{H}^{\mathrm{m}}\right)^T \end{bmatrix}, \tag{48}$$

$$= \begin{bmatrix} \boldsymbol{L}^{\mathrm{m}}(\lambda) \\ \boldsymbol{K} \boldsymbol{L}^{\mathrm{m}}(\lambda) \end{bmatrix}, \tag{49}$$

where $\boldsymbol{K} = \boldsymbol{\Sigma}^{\mathrm{um}} \left(\boldsymbol{\Sigma}^{\mathrm{mm}}\right)^{-1}$. This block structure immediately implies that both the drift and diffusion functions separate in the desired manner:

$$\mathbf{d}(\mathbf{X}(\lambda), \lambda) = \begin{bmatrix} -\boldsymbol{L}^{\mathrm{m}}(\lambda) \boldsymbol{R}^{-1} \left(\boldsymbol{H}^{\mathrm{m}}\, \mathbf{X}^{\mathrm{m}}(\lambda) - \boldsymbol{y}\right) \\ -\boldsymbol{K} \boldsymbol{L}^{\mathrm{m}}(\lambda) \boldsymbol{R}^{-1} \left(\boldsymbol{H}^{\mathrm{m}}\, \mathbf{X}^{\mathrm{m}}(\lambda) - \boldsymbol{y}\right) \end{bmatrix} = \begin{bmatrix} \mathbf{d}^{\mathrm{m}}(\mathbf{X}^{\mathrm{m}}(\lambda), \lambda) \\ \boldsymbol{\Sigma}^{\mathrm{um}} \left(\boldsymbol{\Sigma}^{\mathrm{mm}}\right)^{-1} \mathbf{d}^{\mathrm{m}}(\mathbf{X}^{\mathrm{m}}(\lambda), \lambda) \end{bmatrix}, \tag{50}$$

$$\mathbf{G}(\mathbf{X}(\lambda), \lambda) = \begin{bmatrix} \boldsymbol{L}^{\mathrm{m}}(\lambda) \boldsymbol{R}^{-1/2} \\ \boldsymbol{K} \boldsymbol{L}^{\mathrm{m}}(\lambda) \boldsymbol{R}^{-1/2} \end{bmatrix} = \begin{bmatrix} \mathbf{G}^{\mathrm{m}}(\mathbf{X}^{\mathrm{m}}(\lambda), \lambda) \\ \boldsymbol{\Sigma}^{\mathrm{um}} \left(\boldsymbol{\Sigma}^{\mathrm{mm}}\right)^{-1} \mathbf{G}^{\mathrm{m}}(\mathbf{X}^{\mathrm{m}}(\lambda), \lambda) \end{bmatrix}, \tag{51}$$

confirming the validity of eqs. (26) and (27).

Burnished flow filter

The burnished flow filter was recently derived by constructing an SDE whose solution matches the Kalman update for linear systems. The drift and diffusion functions for this flow are given by:[21]

$$\mathbf{d}(\mathbf{X}(\lambda), \lambda) = \boldsymbol{A}\boldsymbol{M}\left(\boldsymbol{H}\,\mathbf{X}(\lambda) - \boldsymbol{y}\right), \tag{52}$$

$$\mathbf{G}(\mathbf{X}(\lambda), \lambda) = \exp(\boldsymbol{A}\,(\lambda - 1))\,\boldsymbol{L}\boldsymbol{R}^{1/2}, \tag{53}$$

where

$$\boldsymbol{L} = \boldsymbol{\Sigma}\boldsymbol{H}^T\left(\boldsymbol{H}\boldsymbol{\Sigma}\boldsymbol{H}^T + \boldsymbol{R}\right)^{-1}, \tag{54}$$

$$\boldsymbol{A} = \log(\boldsymbol{I} - \boldsymbol{L}\boldsymbol{H}), \tag{55}$$

$$\boldsymbol{M} = \boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{H}^T)^{-1}. \tag{56}$$

Consider the partitioned forms of the prior covariance matrix and measurement matrix from eq. (32), together with the definition of the innovation covariance $\boldsymbol{S} = \boldsymbol{S}_{\lambda=1}$ from eq. (33). Substituting these expressions yields

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{\Sigma}^{\mathrm{mm}}\,(\boldsymbol{H}^{\mathrm{m}})^T\,\boldsymbol{S}^{-1} \\ \boldsymbol{\Sigma}^{\mathrm{um}}\,(\boldsymbol{H}^{\mathrm{m}})^T\,\boldsymbol{S}^{-1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{L}^{\mathrm{m}} \\ \boldsymbol{K}\boldsymbol{L}^{\mathrm{m}} \end{bmatrix}, \tag{57}$$

where $\boldsymbol{K} = \boldsymbol{\Sigma}^{\mathrm{um}}\,(\boldsymbol{\Sigma}^{\mathrm{mm}})^{-1}$. With this block structure in mind, the matrices $\boldsymbol{A}$ and $\boldsymbol{M}$ can be manipulated as follows. First, consider $\boldsymbol{A}$:

$$\begin{aligned}
\boldsymbol{A} &= \log(\boldsymbol{I} - \boldsymbol{L}\boldsymbol{H}), && (58)\\
&= \log\left(\begin{bmatrix} \boldsymbol{I} - \boldsymbol{L}^{\mathrm{m}}\boldsymbol{H}^{\mathrm{m}} & \mathbf{0} \\ -\boldsymbol{K}\boldsymbol{L}^{\mathrm{m}}\boldsymbol{H}^{\mathrm{m}} & \boldsymbol{I} \end{bmatrix}\right), && (59)\\
&= \begin{bmatrix} -\boldsymbol{L}^{\mathrm{m}}\boldsymbol{H}^{\mathrm{m}} & \mathbf{0} \\ -\boldsymbol{K}\boldsymbol{L}^{\mathrm{m}}\boldsymbol{H}^{\mathrm{m}} & \mathbf{0} \end{bmatrix} - \frac{1}{2}\begin{bmatrix} -\boldsymbol{L}^{\mathrm{m}}\boldsymbol{H}^{\mathrm{m}} & \mathbf{0} \\ -\boldsymbol{K}\boldsymbol{L}^{\mathrm{m}}\boldsymbol{H}^{\mathrm{m}} & \mathbf{0} \end{bmatrix}^2 + \frac{1}{3}\begin{bmatrix} -\boldsymbol{L}^{\mathrm{m}}\boldsymbol{H}^{\mathrm{m}} & \mathbf{0} \\ -\boldsymbol{K}\boldsymbol{L}^{\mathrm{m}}\boldsymbol{H}^{\mathrm{m}} & \mathbf{0} \end{bmatrix}^3 - \dots, && (60)\\
&= \begin{bmatrix} -\boldsymbol{L}^{\mathrm{m}}\boldsymbol{H}^{\mathrm{m}} & \mathbf{0} \\ -\boldsymbol{K}\boldsymbol{L}^{\mathrm{m}}\boldsymbol{H}^{\mathrm{m}} & \mathbf{0} \end{bmatrix} - \frac{1}{2}\begin{bmatrix} (-\boldsymbol{L}^{\mathrm{m}}\boldsymbol{H}^{\mathrm{m}})^2 & \mathbf{0} \\ \boldsymbol{K}\,(-\boldsymbol{L}^{\mathrm{m}}\boldsymbol{H}^{\mathrm{m}})^2 & \mathbf{0} \end{bmatrix} + \frac{1}{3}\begin{bmatrix} (-\boldsymbol{L}^{\mathrm{m}}\boldsymbol{H}^{\mathrm{m}})^3 & \mathbf{0} \\ \boldsymbol{K}\,(-\boldsymbol{L}^{\mathrm{m}}\boldsymbol{H}^{\mathrm{m}})^3 & \mathbf{0} \end{bmatrix} - \dots, && (61)\\
&= \begin{bmatrix} \log(\boldsymbol{I} - \boldsymbol{L}^{\mathrm{m}}\boldsymbol{H}^{\mathrm{m}}) & \mathbf{0} \\ \boldsymbol{K}\log(\boldsymbol{I} - \boldsymbol{L}^{\mathrm{m}}\boldsymbol{H}^{\mathrm{m}}) & \mathbf{0} \end{bmatrix}, && (62)\\
&= \begin{bmatrix} \boldsymbol{A}^{\mathrm{m}} & \mathbf{0} \\ \boldsymbol{K}\boldsymbol{A}^{\mathrm{m}} & \mathbf{0} \end{bmatrix}. && (63)
\end{aligned}$$

Similarly, for $\boldsymbol{M}$:

$$\begin{aligned}
\boldsymbol{M} &= \boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{H}^T)^{-1}, && (64)\\
&= \begin{bmatrix} (\boldsymbol{H}^{\mathrm{m}})^T\,(\boldsymbol{H}^{\mathrm{m}}\,(\boldsymbol{H}^{\mathrm{m}})^T)^{-1} \\ \mathbf{0} \end{bmatrix}. && (65)
\end{aligned}$$

Combining these results, the drift function becomes

$$\begin{aligned}
\mathbf{d}(\mathbf{X}(\lambda), \lambda) &= \begin{bmatrix} \boldsymbol{A}^{\mathrm{m}}\,(\boldsymbol{H}^{\mathrm{m}})^T\,(\boldsymbol{H}^{\mathrm{m}}\,(\boldsymbol{H}^{\mathrm{m}})^T)^{-1}\,(\boldsymbol{H}^{\mathrm{m}}\mathbf{X}^{\mathrm{m}} - \boldsymbol{y}) \\ \boldsymbol{K}\boldsymbol{A}^{\mathrm{m}}\,(\boldsymbol{H}^{\mathrm{m}})^T\,(\boldsymbol{H}^{\mathrm{m}}\,(\boldsymbol{H}^{\mathrm{m}})^T)^{-1}\,(\boldsymbol{H}^{\mathrm{m}}\mathbf{X}^{\mathrm{m}} - \boldsymbol{y}) \end{bmatrix}, && (66)\\
&= \begin{bmatrix} \mathbf{d}^{\mathrm{m}}(\mathbf{X}^{\mathrm{m}}(\lambda), \lambda) \\ \boldsymbol{\Sigma}^{\mathrm{um}}\,(\boldsymbol{\Sigma}^{\mathrm{mm}})^{-1}\,\mathbf{d}^{\mathrm{m}}(\mathbf{X}^{\mathrm{m}}(\lambda), \lambda) \end{bmatrix}, && (67)
\end{aligned}$$

which satisfies eq. (26). For the diffusion term, consider now the exponential factor:

$$\boldsymbol{B} = \exp\{\boldsymbol{A}\,(\lambda-1)\}\,, \tag{68}$$

$$= \boldsymbol{I} + \begin{bmatrix} \boldsymbol{A}^{\mathrm{m}}\,(\lambda-1) & \mathbf{0} \\ \boldsymbol{K}\boldsymbol{A}^{\mathrm{m}}\,(\lambda-1) & \mathbf{0} \end{bmatrix} + \frac{1}{2}\begin{bmatrix} \boldsymbol{A}^{\mathrm{m}}\,(\lambda-1) & \mathbf{0} \\ \boldsymbol{K}\boldsymbol{A}^{\mathrm{m}}\,(\lambda-1) & \mathbf{0} \end{bmatrix}^2 + \frac{1}{3}\begin{bmatrix} \boldsymbol{A}^{\mathrm{m}}\,(\lambda-1) & \mathbf{0} \\ \boldsymbol{K}\boldsymbol{A}^{\mathrm{m}}\,(\lambda-1) & \mathbf{0} \end{bmatrix}^3 + \dots, \tag{69}$$

$$= \boldsymbol{I} + \begin{bmatrix} (\boldsymbol{A}^{\mathrm{m}}\,(\lambda-1)) & \mathbf{0} \\ \boldsymbol{K}\boldsymbol{A}^{\mathrm{m}}\,(\lambda-1) & \mathbf{0} \end{bmatrix} + \frac{1}{2}\begin{bmatrix} (\boldsymbol{A}^{\mathrm{m}}\,(\lambda-1))^2 & \mathbf{0} \\ \boldsymbol{K}\,(\boldsymbol{A}^{\mathrm{m}}\,(\lambda-1))^2 & \mathbf{0} \end{bmatrix} + \frac{1}{3}\begin{bmatrix} (\boldsymbol{A}^{\mathrm{m}}\,(\lambda-1))^3 & \mathbf{0} \\ \boldsymbol{K}\,(\boldsymbol{A}^{\mathrm{m}}\,(\lambda-1))^3 & \mathbf{0} \end{bmatrix} + \dots, \tag{70}$$

$$= \begin{bmatrix} \exp\{\boldsymbol{A}^{\mathrm{m}}\,(\lambda-1)\} & \mathbf{0} \\ \boldsymbol{K}\exp\{\boldsymbol{A}^{\mathrm{m}}\,(\lambda-1)\} - \boldsymbol{K} & \boldsymbol{I} \end{bmatrix}. \tag{71}$$

Substituting this expression along with the block form of $\boldsymbol{L}$ leads to

$$\mathbf{G}(\mathbf{X}(\lambda),\lambda) = \boldsymbol{B}\boldsymbol{L}\boldsymbol{R}^{1/2}, \tag{72}$$

$$= \begin{bmatrix} \exp\{\boldsymbol{A}^{\mathrm{m}}\,(\lambda-1)\} & \mathbf{0} \\ \boldsymbol{K}\exp\{\boldsymbol{A}^{\mathrm{m}}\,(\lambda-1)\} - \boldsymbol{K} & \boldsymbol{I} \end{bmatrix}\begin{bmatrix} \boldsymbol{L}^{\mathrm{m}} \\ \boldsymbol{K}\boldsymbol{L}^{\mathrm{m}} \end{bmatrix}\boldsymbol{R}^{1/2}, \tag{73}$$

$$= \begin{bmatrix} \exp\{\boldsymbol{A}^{\mathrm{m}}\,(\lambda-1)\}\,\boldsymbol{L}^{\mathrm{m}} \\ \boldsymbol{K}\exp\{\boldsymbol{A}^{\mathrm{m}}\,(\lambda-1)\}\,\boldsymbol{L}^{\mathrm{m}} \end{bmatrix}\boldsymbol{R}^{1/2} \tag{74}$$

$$= \begin{bmatrix} \mathbf{G}^{\mathrm{m}}(\mathbf{X}^{\mathrm{m}}(\lambda),\lambda)\,, \\ \boldsymbol{\Sigma}^{\mathrm{um}}\,(\boldsymbol{\Sigma}^{\mathrm{mm}})^{-1}\,\mathbf{G}^{\mathrm{m}}(\mathbf{X}^{\mathrm{m}}(\lambda),\lambda) \end{bmatrix}, \tag{75}$$

which satisfies eq. (27).

NUMERICAL EXAMPLE

The goal of this numerical example is to demonstrate that the proposed method improves the computational efficiency of Gaussian particle flow filters without compromising estimation accuracy. As a benchmark problem, we consider the Lorenz '96 system, a standard test case for data assimilation and forecasting methods. The system is governed by a set of coupled ordinary differential equations that model the advection, dissipation, and external forcing of an atmospheric-like quantity. For a n-dimensional system, the dynamics are defined as follows:

$$\frac{d\,x_i(t)}{dt} = (x_{i+1}(t) - x_{i-2}(t))\,x_{i-1}(t) - x_i(t) + F, \qquad i = 1,2,\dots n, \tag{76}$$

with cyclic boundary conditions $x_{-1}(t) = x_{n-1}(t)$, $x_0(t) = x_n(t)$, $x_{n+1}(t) = x_1(t)$, and forcing parameter $F = 10$, which induces chaotic behavior. In this work, a 40-dimensional state vector is considered, a classical and widely adopted configuration for this problem. To further increase the level of difficulty, the true trajectory is perturbed by a discrete-time process noise with covariance matrix $\boldsymbol{Q} = \boldsymbol{I}_{n\times n}$.

The nonlinear observation model is defined element-wise as

$$h(x_i(t)) = \frac{x_i(t)}{2}\left[1 + \left(\frac{|x_i(t)|}{f}\right)^{\gamma-1}\right], \tag{77}$$

where γ controls the degree of nonlinearity in the observations and f rescales the magnitude of the measurements. One state variable out of every five is directly observed. The resulting observation operator can therefore be written as

$$\boldsymbol{y}(t) = \left[h(x_1(t)), \quad h(x_6(t)), \quad h(x_{11}(t)), \quad \ldots, \quad h(x_{36}(t))\right]^T + \boldsymbol{\eta}, \tag{78}$$

where

$$\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R}). \tag{79}$$

The three particle flow filters considered in this work, the exact flow filter, the Gromov flow filter, and the burnished flow filter, were implemented both in their standard formulations and in the proposed computationally improved version. All methods were evaluated over 1008 independent Monte Carlo runs. Measurements were assimilated every $\Delta t = 0.2$ time units, corresponding approximately to one model day. Each simulation was carried out for a total duration of 10 time units.

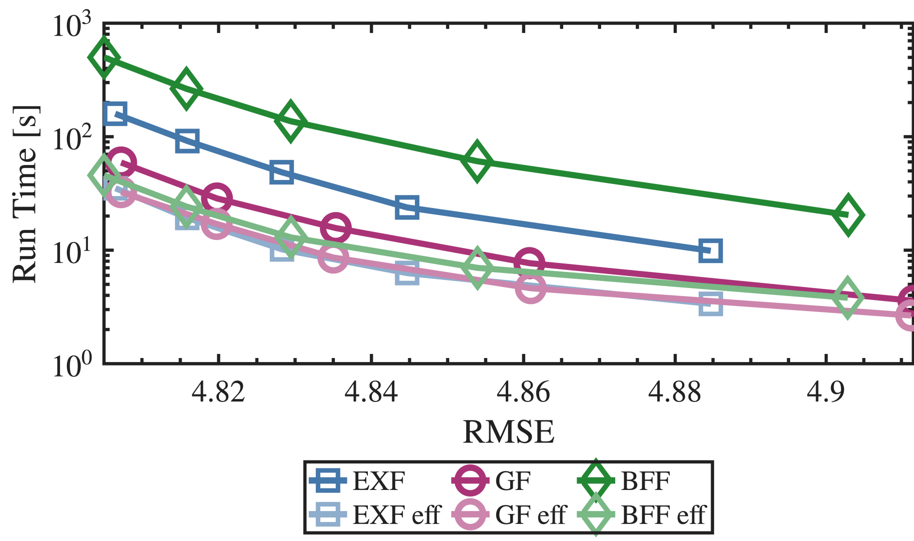

Figure 1. Run time as a function of the RMSE for the three Gaussian particle flow filters considered. Each marker corresponds to a different number of samples.

To compare the different approaches, we used the time-averaged root mean square error (RMSE) and the time-averaged scaled normalized estimation error squared (SNEES).

The time-averaged RMSE is used to assess estimation accuracy and is defined as

$$\text{RMSE} = \frac{1}{N_{\text{t}} N_{\text{m}}} \sum_{k=1}^{N_{\text{t}}} \sum_{j=1}^{N_{\text{m}}} \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(\hat{x}_{k,j}^{(i)} - x_{k,j}^{(i)}\right)^2} \tag{80}$$

where N_{t} is the number of time steps, N_{m} is the number of Monte Carlo runs, $\hat{x}_{k,j}^{(i)}$ is the estimate of the ith state component at time step k in Monte Carlo run j, and $x_{k,j}^{(i)}$ is the corresponding true state. The time-averaged SNEES evaluates statistical consistency and is defined as

$$\text{SNEES} = \frac{1}{N_{\text{t}} N_{\text{m}}} \sum_{k=1}^{N_{\text{t}}} \sum_{j=1}^{N_{\text{m}}} \left(\hat{\boldsymbol{x}}_{k,j} - \boldsymbol{x}_{k,j}\right)^T \boldsymbol{P}_{k,j}^{-1} \left(\hat{\boldsymbol{x}}_{k,j} - \boldsymbol{x}_{k,j}\right) \tag{81}$$

where $\boldsymbol{P}_{k,j}$ is the estimated error covariance at time step k in Monte Carlo run j. Ideally, a SNEES value of 1 indicates a statistically consistent filter. A value greater than one indicates that the filter is overconfident in its uncertainty estimates, whereas a value below one suggests that the filter is conservative.

All filters were carefully optimized to maximize run time efficiency in order to ensure fair computational comparisons. Simulations were executed in MATLAB R2025b on a MacBook Pro equipped with an Apple M4 Pro chip and 24 GB of RAM.

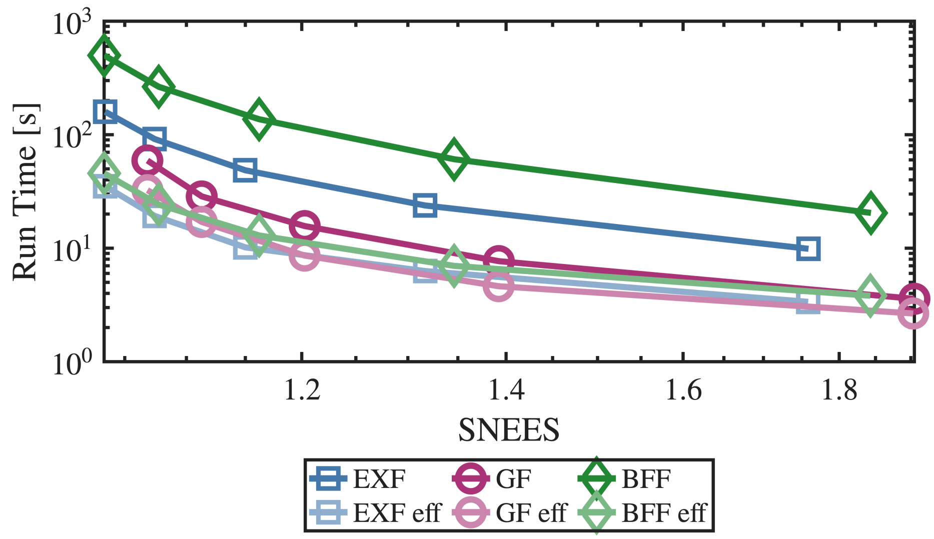

Figure 2. Run time as a function of the SNEES for the three Gaussian particle flow filters considered. Each marker corresponds to a different number of samples.

The Monte Carlo simulations were repeated using a number of samples ranging from 100 to 1000. Figure 1 reports the run time as a function of the RMSE. Each marker in the figure corresponds to a different number of samples. As expected, increasing the number of samples leads to higher computational cost for all filters, while the RMSE decreases, indicating improved estimation accuracy. For a fixed number of samples, no significant differences in RMSE can be observed between the standard implementations and the proposed one. This confirms that the proposed approach preserves estimation accuracy. In contrast, the computational time is reduced significantly across all sample counts and for all filters when using the proposed implementation. For larger sample sizes, the computational time is reduced by up to one order of magnitude.

Figure 2 presents the same analysis in terms of the SNEES. A similar trend can be observed: as the number of samples increases, all filters exhibit improved statistical consistency. The exact flow filter tends to be slightly overconfident, whereas the Gromov and burnished flow filters appear mildly conservative. Again, for a fixed number of samples, the SNEES values obtained with the proposed implementation are indistinguishable from those of the standard implementation. However, the run time is consistently reduced.

CONCLUSIONS

This work presented an efficient implementation of Gaussian particle flow filters for nonlinear estimation problems in which only a subset of the state enters the measurement function. By exploiting this structure, the proposed approach reduces the cost of pseudo-time flow integration by flowing only the measured subspace and then updating the remaining states through a conditional Gaussian update, while preserving the estimator's statistical behavior.

We established theoretical equivalence between the proposed implementation and several widely used Gaussian particle flow filters under the stated conditions, showing that the reduced-order integration yields the same posterior ensemble as the standard full-state flow. Numerical simulations confirmed that the method achieves essentially identical estimation accuracy and consistency metrics compared to standard implementations, while substantially reducing run time, up to an order of magnitude for large ensemble sizes.

REFERENCES

[1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, 1960.

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

[3] S. J. Julier and J. K. Uhlmann, "New extension of the Kalman filter to nonlinear systems," *Signal processing, sensor fusion, and target recognition VI*, Vol. 3068, Spie, 1997, pp. 182–193.

[4] N. J. Gordon, D. J. Salmond, and A. F. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE proceedings F (radar and signal processing)*, Vol. 140, IET, 1993, pp. 107–113.

[5] A. Doucet, N. De Freitas, and N. Gordon, "An introduction to sequential Monte Carlo methods," *Sequential Monte Carlo methods in practice*, pp. 3–14, Springer, 2001.

[6] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *Journal of the American statistical association*, Vol. 94, No. 446, 1999, pp. 590–599.

[7] R. Van Der Merwe, A. Doucet, N. De Freitas, and E. Wan, "The unscented particle filter," *Advances in neural information processing systems*, Vol. 13, 2000.

[8] F. Daum and J. Huang, "Nonlinear filters with particle flow," *Signal and Data Processing of Small Targets 2009*, Vol. 7445, SPIE, 2009, pp. 315–323.

[9] 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, 2015, pp. 752–760.

[10] F. Giraldo-Grueso, A. A. Popov, and R. Zanetti, "Gaussian Mixture-Based Point Mass Filtering With Applications to Terrain-Relative Navigation," *IEEE Transactions on Aerospace and Electronic Systems*, 2025.

[11] J. Wang, L. Zhao, and X. Su, "Marginalized particle flow filter," *Circuits, Systems, and Signal Processing*, Vol. 38, No. 7, 2019, pp. 3152–3169.

[12] H. W. Sorenson and D. L. Alspach, "Recursive Bayesian estimation using Gaussian sums," *Automatica*, Vol. 7, 1971, pp. 465–479.

[13] 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.

[14] K. J. Craft and K. J. DeMars, "Stein variational gradient descent for non-Bayesian particle flow," *2023 26th International Conference on Information Fusion (FUSION)*, IEEE, 2023, pp. 1–8.

[15] D. Prossel and U. D. Hanebeck, "Progressive particle filtering using projected cumulative distributions," *2023 IEEE Symposium Sensor Data Fusion and International Conference on Multisensor Fusion and Integration (SDF-MFI)*, IEEE, 2023, pp. 1–8.

[16] C.-C. Hu and P. J. Van Leeuwen, "A particle flow filter for high-dimensional system applications," *Quarterly Journal of the Royal Meteorological Society*, Vol. 147, No. 737, 2021, pp. 2352–2374.

[17] F. Daum and J. Huang, "a baker's dozen of new particle flows for nonlinear filters, Bayesian decisions and transport," *Signal Processing, Sensor/Information Fusion, and Target Recognition XXIV*, Vol. 9474, SPIE, 2015, pp. 176–188.

[18] A. H. Jazwinski, *Stochastic processes and filtering theory*. Courier Corporation, 2007.

[19] F. Daum, J. Huang, and A. Noushin, "Exact particle flow for nonlinear filters," *Signal processing, sensor fusion, and target recognition XIX*, Vol. 7697, SPIE, 2010, pp. 92–110.

[20] F. Daum, J. Huang, and A. Noushin, "Generalized Gromov method for stochastic particle flow filters," *Signal Processing, Sensor/Information Fusion, and Target Recognition XXVI*, Vol. 10200, SPIE, 2017, pp. 159–171.

[21] R. Mamich, K. Michaelson, A. A. Popov, and R. Zanetti, "Burnished Flow Filter," *2024 27th International Conference on Information Fusion (FUSION)*, IEEE, 2024, pp. 1–8.

[22] M. A. Woodbury, *Inverting modified matrices*. Department of Statistics, Princeton University, 1950.