Burnished Flow Filter

Rachel Mamich*, Kristen Michaelson*, Andrey A. Popov† and Renato Zanetti*†
*Dept. of Aerospace Engineering: The University of Texas at Austin, Austin, Texas
Email: rachel.mamich@utexas.edu, kmichaelson@utexas.edu, renato@utexas.edu
†The Oden Institute for Computational Engineering and Sciences: The University of Texas at Austin, Austin, Texas
Email: andrey.a.popov@utexas.edu

***Abstract*—The Burnished Flow Filter is a particle flow filter constructed from the Kalman filter measurement update equations. The derivation for this filter begins by assuming the classic Kalman Filter measurement update equations are the solution to a stochastic differential equation. By using these well known equations, the derivation of this filter follows naturally to an engineer with a Kalman filtering background. The work presented here shows the derivation, and application of this filter on both linear and nonlinear problems. The Burnished Flow Filter is benchmarked against the widely used Gromov Flow Filter, revealing similar performance in linear problems and demonstrating superior consistency in the nonlinear scenarios under study. Additionally, the Burnished Flow Filter exhibits a smoother flow compared to the Gromov Flow Filter, as evidenced by a smaller state update during the first substep of the measurement update.**

***Index Terms*—particle flow, SDE, nonlinear filtering**

I. Introduction

The Kalman Filter (KF) provides the optimal solution to a linear system [1], [2]. However, for nonlinear systems, the KF cannot guarantee optimal results, necessitating specialized approaches. One such adaptation is the Extended Kalman Filter (EKF), which uses a first-order Taylor series approximation and is widely used in industry to address nonlinear challenges[3]. Despite its popularity, the EKF has notable limitations due to its dependence on Jacobians calculated along its reference trajectory [4].

Nonlinear systems often exhibit complex, unintuitive behavior that requires nonlinear filters. As a result, nonlinear filtering has been the focus of numerous studies [5]. One major class of filters that has gained traction for handling highly nonlinear problems is particle flow filters. These filters draw a set of random samples from the prior distribution and solve a differential equation, with an initial condition at each particle, to 'flow' the samples toward the posterior distribution. As the number of particles increases, the sample PDF is expected to converge toward the posterior PDF [6].

Over the years, several particle flow filters have been proposed, starting with the work of [7]. Subsequent particle flow studies have fallen into two categories: deterministic (zero-diffusion) particle flow [8]–[11] and stochastic particle flow [12]–[14]. Deterministic particle flow filters utilize an Ordinary Differential Equation (ODE) to flow the particles from the prior to the posterior distribution whereas stochastic particle flow employs a Stochastic Differential Equation (SDE) [15]. In [16] Dai and Daum analyze the relationship between deterministic and stochastic flows in an effort to improve the transient dynamics of particle flows. One noted challenge in particle flow filtering mentioned in Dai and Daum's work is that the governing differential equations of particle flow are stiff.

This work introduces a novel approach to particle flow filtering: the Burnished Flow Filter (BFF). The BFF is a new stochastic particle flow algorithm that combines the power of particle flow with the familiarity of the KF update equations, yielding a capable and accessible algorithm. Previous work on stochastic flow finds a solution to the SDE by applying the Fokker-Planck equation in an intricate series of manipulations [15]. In contrast, the BFF first assumes that the solution to the SDE takes a familiar form in the KF measurement update equations and backfills the details accordingly.

This paper is organized as follows: in Section II, background on previous work is presented. In Section III, the derivation of the BFF equations is shown for linear systems. Section IV provides additional information for implementing both the BFF and GFF. In Sections V–VII results from simulations ran on a linear system, a two dimensional range observation system, and an Ikeda Map system are presented. Section VIII concludes the major takeaways from this work.

II. Background

Consider the following system of equations that describe the dynamics of a state vector $\mathbf{x}$ and measurements of that state vector $\mathbf{y}$:

$$\mathbf{x}_{k+1} = \mathbf{f}\left(\mathbf{x}_k, t\right) + \boldsymbol{\nu}, \quad \mathbf{y} = \mathbf{h}\left(\mathbf{x}\right) + \boldsymbol{\eta}, \tag{1}$$

where t is time, $\mathbf{f}\left(\mathbf{x}_k, t\right)$ is the dynamics function that relates the state and its propagation (linear or nonlinear), $\boldsymbol{\nu}$ is the process noise in the system, $\mathbf{h}\left(\mathbf{x}\right)$ is the measurement function (linear or nonlinear), and $\boldsymbol{\eta}$ is the measurement noise. In this work it is assumed that $\boldsymbol{\nu}$ and $\boldsymbol{\eta}$ are zero mean Gaussian random vectors vectors with covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ respectively.

State estimation aims to provide an estimate of the true state and its corresponding PDF. In this paper, the true state is denoted as $\mathbf{x}$, while the filter's estimate of the true state is represented as $\hat{\mathbf{x}}$. Filters generally achieve this through a two-step process: propagation and measurement update.

This work was supported by a NASA Space Technology Graduate Research Opportunity, NASA grant number: 80NSSC22K1204.

The EKF makes the assumption that the state distribution is characterized by its first two moments. This is often considered a 'Gaussian' assumption because a Gaussian probability density function (PDF) is fully defined by its mean and covariance [17], although this isn't strictly necessary. During propagation, the EKF propagates the state without noise and simply adds the process noise matrix to the current covariance estimate,

$$\begin{aligned} \hat{\mathbf{x}}_{k+1}^{-} &= \mathbf{f}(\hat{\mathbf{x}}_{k}^{+}, t), \quad \mathbf{P}_{k+1}^{-} = \mathbf{F}_k \mathbf{P}_k^{-} \mathbf{F}_k^{T} + \mathbf{Q}, \\ \mathbf{F}_k &= \left. \frac{\partial \mathbf{x}_{k+1}}{\partial \mathbf{x}_k} \right|_{\hat{\mathbf{x}}_k^{+}}, \end{aligned} \tag{2}$$

where the superscript $()^{+}$ indicates the variable is the posterior estimate, the superscript $()^{-}$ indicates the variable is the prior estimate, the subscript $()_k$ indicates the variable is represented at the discrete time step k, the subscript $()_{k+1}$ indicates the variable is represented at time step $k+1$, $\mathbf{P}_k$ is the covariance matrix of the state, and $\mathbf{F}_k$ is the discrete time state transition matrix evaluated on $\hat{\mathbf{x}}_k^{+}$. In an EKF the measurement update is performed in a single step based on the measurement Jacobian at the current best estimate:

$$\begin{aligned} \boldsymbol{K}_{k+1} &= \boldsymbol{P}_{k+1}^{-} \boldsymbol{H}_{k+1}^{T} \left(\boldsymbol{H}_{k+1} \boldsymbol{P}_{k+1}^{-} \boldsymbol{H}_{k+1}^{T} + \boldsymbol{R} \right)^{-1}, \\ \hat{\boldsymbol{x}}_{k+1}^{+} &= (\boldsymbol{I} - \boldsymbol{K}_{k+1} \boldsymbol{H}_{k+1}) \hat{\boldsymbol{x}}_{k+1}^{-} + \boldsymbol{K}_{k+1} \boldsymbol{y}_{k+1}, \\ \boldsymbol{P}_{k+1}^{+} &= (\boldsymbol{I} - \boldsymbol{K}_{k+1} \boldsymbol{H}_{k+1}) \boldsymbol{P}_{k+1}^{-} (\boldsymbol{I} - \boldsymbol{K}_{k+1} \boldsymbol{H}_{k+1})^{T} + \\ & \quad \boldsymbol{K}_{k+1} \boldsymbol{R} \boldsymbol{K}_{k+1}^{T}, \\ \mathbf{H}_{k+1} &= \left. \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k+1}^{-}}, \end{aligned} \tag{3}$$

where $\mathbf{H}$ is the measurement Jacobian evaluated on $\hat{\mathbf{x}}_{k+1}^{-}$. It should be noted that the covariance update shown in (3) is in the Joseph form [3].

For initially Gaussian, linear systems with Gaussian additive noise, the system will remain Gaussian. The EKF makes the assumption that the system is 'linear enough' about the state estimate such that the errors in the state are essentially linear. This assumption implies that the PDF will continue to be characterized by its first two moments through time propagation and measurement updates. However, for highly nonlinear systems, Jacobians can change rapidly, compromising this 'linear enough' assumption. As a result, the EKF is susceptible to significant nonlinearities.

Particle flow filters on the other hand, use a collection of random samples in order to estimate the full PDF as shown in Figure 1, step 1. Particle flow filters propagate the dynamics of the system by applying (1) to each particle [15]. This can be seen in Figure 1, step 2. Here, each particle is 'nudged' with its own random sample(s) from the process noise distribution. This differs from the standard EKF where the noise term is ignored during state propagation. This practice of 'nudging' particles is not unique to particle flow filters; common particle filters like the bootstrap particle filter also apply random samples of process noise to each particle during propagation [6]. Another filter that leverages the same propagation and a different measurement update is the Ensemble Gaussian

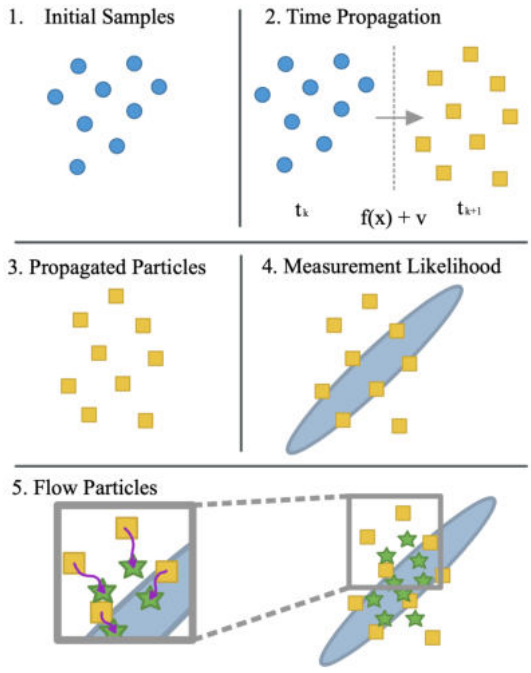

Fig. 1. A visual representation of how particles are processed in a particle flow algorithm. Beginning with initial samples, the samples are propagated through time, then the particles are flowed towards the posterior distribution informed by the measurement likelihood. This cycle repeats until there are no more measurements to process.

Mixture (EnGMF) [18]–[20]. While the propagation is identical, lending itself to classifying an EnGMF as a particle filter, the measurement update relies on performing kernel density estimation followed by a local Gaussian update on each particle, resulting in an approximation of the posterior by a Gaussian mixture model.

Although the propagation step is identical for both particle filters and particle flow filters, the latter can perform well in the absence of process noise while the former generally struggle. When noise levels are low, particle filters can be subject to particle degeneracy, a condition where only a few samples have most of the weight [21]. Since particle flow filters weight each particle equally, there is little risk of degeneracy and thus these filters are more robust to low noise levels.

Once the particles are all propagated, the measurement update is applied to each one. This is where particle flow filters differ from particle filters, and where particle flow filters tend to differ from one another. A common feature across Bayesian estimation is the leveraging of the measurement likelihood $p_{\mathbf{y}|\mathbf{x}}$ (Figure 1, step 4) to update or 'flow' the filter estimate to better represent the posterior distribution given the measurement. The flow of particles is defined via a SDE (Figure 1, step 5). This SDE takes the form [15]:

$$d\mathbf{x} = \mathbf{g}\left(\mathbf{x}, \lambda, \mathbf{y}\right) d\lambda + \mathbf{B}_{\lambda}\left(\mathbf{x}, \lambda\right) d\mathbf{w}_{\lambda}, \tag{4}$$

where λ is a pseudotime variable, $\mathbf{g}(\mathbf{x}, \lambda, \mathbf{y})$ is the drift term, $\mathbf{y}$ is deterministic (i.e., it is the outcome of a random variable, not the random variable itself), $\mathbf{B}_{\lambda}$ is the diffusion term, and

$d\mathbf{w}_\lambda$ is a Weiner process. The diffusion term is set to zero for deterministic flow.

Given a number of convenient linearity and Gaussianity assumptions, the Gromov flow drift and diffusion terms can be expressed in the following simplified form [22], [23]:

$$\begin{aligned}
&\mathbf{g}\left(\mathbf{x},\lambda,\mathbf{y}\right)=\left(\mathbf{P}^{-1}+\lambda\mathbf{H}^{T}\mathbf{R}^{-1}\mathbf{H}\right)^{-1}\mathbf{H}^{T}\mathbf{R}^{-1}\left(\mathbf{y}-\mathbf{h}(\mathbf{x})\right),\\
&\mathbf{Q}_\lambda=\left(\mathbf{P}^{-1}+\lambda\mathbf{H}^{T}\mathbf{R}^{-1}\mathbf{H}\right)^{-1}\mathbf{H}^{T}\mathbf{R}^{-1}\mathbf{H}\left(\sim\right)^{-1},\\
&\mathbf{Q}_\lambda=\mathbf{B}_\lambda\mathbf{B}_\lambda^{T}=\mathbf{U}\mathbf{S}\mathbf{V}^{T},
\end{aligned}\tag{5}$$

where $(\sim)$ indicates that the quantity in the parentheses is the same as the previous set of parentheses in the equation, and $\mathbf{Q}_\lambda$ is an intermediate matrix used to solve for $\mathbf{B}_\lambda$. For clarity, the variables $\mathbf{P},\mathbf{H}$, and $\mathbf{R}$ in (5) are the same as those in the Kalman filter measurement update equations in (3). When $\mathbf{Q}_\lambda$ is not singular, $\mathbf{B}_\lambda=\sqrt{\mathbf{Q}_\lambda}$. Otherwise $\mathbf{B}_\lambda$ may be found using singular value decomposition of $\mathbf{Q}_\lambda$ as $\mathbf{B}_\lambda=\mathbf{U}\sqrt{\mathbf{S}}\mathbf{V}^{T}$ where the $\sqrt{\mathbf{S}}$ is the component-wise square root of the matrix $\mathbf{S}$.

The burnished flow filter (BFF) introduced in this work is compared against the Gromov flow filter. The SDE governing Gromov flow provides the exact Bayesian posterior in the linear/Gaussian measurement case. More information on Gromov flow can be found in [12], [15], [22], [24]. The next section will cover the derivation of the BFF, which is also exact for linear/Gaussian measurements.

III. Derivation of the BFF Measurement Update Equations for a Linear Measurement

The EKF is one of the most commonly used filters for state estimation. The intention in designing the BFF is to create a particle flow filter that is both powerful and easy to understand. Consider the linear measurement,

$$\boldsymbol{y}=\boldsymbol{H}\boldsymbol{x}(t)+\boldsymbol{\eta},\tag{6}$$

where again, the measurement noise $\boldsymbol{\eta}$ is zero mean and Gaussian. Assuming $\boldsymbol{x}(0)\sim\mathcal{N}(\hat{\boldsymbol{x}}_0,\boldsymbol{P}_0)$, the posterior distribution is also Gaussian with mean and covariance given by the KF update (3). It should be noted that it is assumed here that the random variables are Gaussian distributed, but this is not necessary for the derivation. Now, assume the following SDE,

$$\begin{aligned}
&d\boldsymbol{x}(\lambda)=\left(\boldsymbol{A}(\lambda)\boldsymbol{x}(\lambda)+\boldsymbol{B}(\lambda)\boldsymbol{y}\right)d\lambda+\boldsymbol{C}(\lambda)d\mathbf{w}_\lambda(\lambda),\\
&\boldsymbol{x}(0)\sim\mathcal{N}(\hat{\boldsymbol{x}}_0,\boldsymbol{P}_0),\\
&\quad\lambda\in[0,1],
\end{aligned}\tag{7}$$

where again $\boldsymbol{y}$ is deterministic, and $d\mathbf{w}_\lambda$ is zero mean white noise with power spectral density (PSD) $\boldsymbol{I}$. Note that given the choices of $\mathbf{A}_\lambda$ and $\mathbf{B}_\lambda$ in (7), a linear relationship is deliberately forced between the stochastic flow and the current state estimate and measurement values. This allows one to derive closed-form expressions for the matrices $\mathbf{A}_\lambda$ and $\mathbf{B}_\lambda$ below.

If one makes the assumption that the state $\boldsymbol{x}(\lambda)$ remains Gaussian at all times, integrating from zero to one, the mean and covariance are given by,

$$\begin{aligned}
\hat{\mathbf{x}}(1)&=\boldsymbol{\Phi}_{\mathbf{A}}(1,0)\;\hat{\mathbf{x}}(0)+\int_0^1\boldsymbol{\Phi}_{\mathbf{A}}(1,\tau)\mathbf{B}(\tau)\;\boldsymbol{y}\;d\tau,\\
\mathbf{P}(1)&=\boldsymbol{\Phi}_{\mathbf{A}}(1,0)\mathbf{P}(0)\boldsymbol{\Phi}_{\mathbf{A}}(1,0)^{T}+\\
&\quad\int_0^1\boldsymbol{\Phi}_{\mathbf{A}}(1,\tau)\mathbf{C}(\tau)\mathbf{C}(\tau)^{T}\boldsymbol{\Phi}_{\mathbf{A}}(1,\tau)^{T}d\tau,
\end{aligned}\tag{8}$$

where $\boldsymbol{\Phi}_{\boldsymbol{A}}(1,\tau)$ is the state transition matrix (STM) of $\boldsymbol{A}$ from τ to 1. As a reminder some useful properties of the STM are:

$$\begin{aligned}
\frac{\partial}{\partial\tau}\boldsymbol{\Phi}_{\mathbf{A}}(1,\tau)&=-\boldsymbol{\Phi}_{\mathbf{A}}(1,\tau)\mathbf{A}(\tau),\\
\boldsymbol{\Phi}_{\mathbf{A}}(0,0)&=\mathbf{I},
\end{aligned}\tag{9}$$

The solution of the SDE (8) matches the Kalman update by,

$$\begin{aligned}
&\boldsymbol{\Phi}_{\boldsymbol{A}}(1,0)=(\boldsymbol{I}-\boldsymbol{K}\boldsymbol{H}),\\
&\int_0^1\boldsymbol{\Phi}_{\boldsymbol{A}}(1,\tau)\boldsymbol{B}(\tau)\;d\tau=\boldsymbol{K},\\
&\int_0^1\boldsymbol{\Phi}_{\boldsymbol{A}}(1,\tau)\boldsymbol{C}(\tau)\boldsymbol{C}(\tau)^{\mathrm{T}}\boldsymbol{\Phi}_{\boldsymbol{A}}(1,\tau)^{\mathrm{T}}d\tau=\boldsymbol{K}\boldsymbol{R}\boldsymbol{K}^{\mathrm{T}}.
\end{aligned}\tag{10}$$

Notice $\boldsymbol{K}$ is a constant, calculating it with the *prior* value of the covariance matrix. There are many possible choices of matrices $\boldsymbol{A}(\lambda)$, $\boldsymbol{B}(\lambda)$, and $\boldsymbol{C}(\lambda)$ that satisfy these equations. Next, intuitive selections for $\boldsymbol{A}(\lambda)$, $\boldsymbol{B}(\lambda)$, and $\boldsymbol{C}(\lambda)$ are provided. A convenient choice for $\boldsymbol{B}(\lambda)$ is,

$$\boldsymbol{B}(\lambda)=-\boldsymbol{A}(\lambda)\boldsymbol{M},\tag{11}$$

where $\mathbf{M}$ is defined later, so that

$$\begin{aligned}
\int_0^1\boldsymbol{\Phi}_{\boldsymbol{A}}(1,\tau)\boldsymbol{B}(\tau)\;d\tau&=-\int_0^1\boldsymbol{\Phi}_{\boldsymbol{A}}(1,\tau)\boldsymbol{A}(\tau)\;d\tau\;\;\boldsymbol{M}\\
&=\int_0^1\frac{\partial}{\partial\tau}\boldsymbol{\Phi}_{\boldsymbol{A}}(1,\tau)\;d\tau\;\;\boldsymbol{M}\\
&=(\boldsymbol{I}-\boldsymbol{\Phi}_{\boldsymbol{A}}(1,0))\;\;\boldsymbol{M}\\
&=\boldsymbol{K}\boldsymbol{H}\boldsymbol{M}.
\end{aligned}\tag{12}$$

Typically the number of states, n, is greater than the number of measurements, m, and $\boldsymbol{H}$ has full row rank. If so, one can choose $\boldsymbol{M}=\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}})^{-1}$ to satisfy the condition of (10). If $n\leq m$, and if the $m\times m$ matrix $\boldsymbol{K}\boldsymbol{H}$ is invertible, one can choose $\boldsymbol{M}=(\boldsymbol{K}\boldsymbol{H})^{-1}\boldsymbol{K}$ to satisfy the condition of (10).

Now select a constant $\boldsymbol{A}$ such that,

$$\boldsymbol{A}=\log(\boldsymbol{I}-\boldsymbol{K}\boldsymbol{H})=\sum_{k=1}^{\infty}\frac{(-1)^{k+1}}{k}(-\boldsymbol{K}\boldsymbol{H})^{k},\tag{13}$$

which one can think of as the pseudotime derivative of the KF update. Redefining the matrix $\boldsymbol{A}$ such that $\boldsymbol{A}=-\tilde{\boldsymbol{A}}\boldsymbol{H}$, hence,

$$\tilde{\boldsymbol{A}}=\sum_{k=1}^{\infty}\frac{(-1)^{k}}{k}(-\boldsymbol{K}\boldsymbol{H})^{k-1}\boldsymbol{K},\tag{14}$$

and, once again choosing $\boldsymbol{M} = \boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}})^{-1}$ or $\boldsymbol{M} = (\boldsymbol{K}\boldsymbol{H})^{-1}\boldsymbol{K}$, whichever exists,

$$\begin{aligned} -\tilde{\boldsymbol{A}}\boldsymbol{H} &= \boldsymbol{A} \\ -\tilde{\boldsymbol{A}}\boldsymbol{H}\boldsymbol{M} &= \boldsymbol{A}\boldsymbol{M} \\ \tilde{\boldsymbol{A}} &= -\boldsymbol{A}\boldsymbol{M} = \boldsymbol{B} \end{aligned} \tag{15}$$

and

$$\begin{aligned} d\boldsymbol{x}(\lambda) &= (-\boldsymbol{B}\boldsymbol{H}\boldsymbol{x}(\lambda) + \boldsymbol{B}\boldsymbol{y})d\lambda + \boldsymbol{C}(\lambda)d\boldsymbol{\nu}(\lambda), \\ \boldsymbol{x}(0) &\sim \mathcal{N}(\hat{\boldsymbol{x}}_0, \boldsymbol{P}_0), \end{aligned} \tag{16}$$

then, finally, choose $\mathbf{C}(\lambda)$ as,

$$\boldsymbol{C}(\lambda) = \boldsymbol{\Phi}_{\boldsymbol{A}}(1,\lambda)^{-1}\boldsymbol{K}\boldsymbol{R}^{1/2}. \tag{17}$$

This is the final variable definition required in (7). In summary, during the BFF measurement update, one integrates the following in λ from 0 to 1:

$$\begin{aligned} d\mathbf{x}(\lambda) &= \mathbf{B}\ (\mathbf{y} - \mathbf{H}\mathbf{x}(\lambda))d\lambda + \mathbf{C}(\lambda)d\boldsymbol{\nu}(\lambda), \\ \mathbf{x}(0) &\sim \mathcal{N}(\hat{\mathbf{x}}_0, \mathbf{P}_0), \\ \mathbf{B} &= -\log(\mathbf{I} - \mathbf{K}\mathbf{H})\ \mathbf{M}, \\ \mathbf{K} &= \mathbf{P}_0\mathbf{H}^T\left(\mathbf{H}\mathbf{P}_0\mathbf{H}^T + \mathbf{R}\right)^{-1}, \\ \mathbf{M} &= \mathbf{H}(\mathbf{H}\mathbf{H}^T)^{-1} \text{ or } = (\mathbf{K}\mathbf{H})^{-1}\mathbf{K}, \\ \mathbf{C}(\lambda) &= e^{\mathbf{A}(\lambda-1)}\mathbf{K}\mathbf{R}^{1/2}. \end{aligned} \tag{18}$$

Written differently for ease of comparison with the GFF, the BFF equations can be written as:

$$\begin{aligned} \mathbf{g}\left(\mathbf{x}, \lambda, \mathbf{y}\right) &= \mathbf{B}\left(\mathbf{y} - \hat{\mathbf{y}}\right), \\ \mathbf{B}_\lambda &= \sqrt{\mathbf{Q}_\lambda} = e^{\boldsymbol{A}(\lambda-1)}\boldsymbol{K}\boldsymbol{R}^{1/2} = \mathbf{C}, \end{aligned} \tag{19}$$

where $\hat{\mathbf{y}} = \mathbf{h}(\hat{\mathbf{x}})$ is the estimated measurement based on the current estimate of the state.

For a nonlinear system, the equations are altered in a similar manner to that of the classical KF to the EKF. Each of the Jacobians listed becomes the Jacobian of the nonlinear system evaluated on the most recent best estimate of the state.

The following section details the implementation of the algorithms for the BFF and GFF.

IV. Implementation Overview

First, a brief discussion of the propagation step. The propagation for the BFF and GFF is handled identically. Each of the systems covered in Sections V–VII is a discrete time system. Each propagation step is treated as a single discrete time step that applies random noise samples to each particle. Once all particles have been propagated, $\hat{\mathbf{x}}^-$ and $\mathbf{P}^-$ are calculated using the particles in (20).

$$\begin{aligned} \hat{\mathbf{x}}_k &= \frac{1}{N_p}\sum_{i=1}^{N_p}\mathbf{x}_k^i, \\ \mathbf{P}_k &= \frac{1}{N_p}\sum_{i=1}^{N_p}(\mathbf{x}_k^i - \hat{\mathbf{x}}_k^i)(\mathbf{x}_k^i - \hat{\mathbf{x}}_k^i)^T, \end{aligned} \tag{20}$$

where $\mathbf{x}_k^i$ is the ith particle of time step k and N_p is the number of particles. In (18), $\mathbf{P}_0 = \mathbf{P}^-$.

In the measurement update step the BFF and GFF have similar mechanics. Both use their solution to the SDE (7) and (4) to perform a fixed number of pseudotime steps, N_λ, via Euler-Maruyama propagation for each particle.

$$\mathbf{x}_{j+1} = \mathbf{x}_j + d\lambda\mathbf{g}(\mathbf{x}_j, \lambda, \mathbf{y}) + \mathbf{B}_\lambda d\mathbf{w}_\lambda, \quad d\lambda = \frac{1}{N_\lambda}, \tag{21}$$

where j represents the current pseudotime step. The GFF uses (5) to define $\mathbf{g}(\mathbf{x}_j, \lambda, \mathbf{y})$ and $\mathbf{B}_\lambda$ while the BFF uses (18). The number of pseudotime steps varies problem to problem in this study, but is fixed for each scenario for ease of comparison.

Lastly, (20) is used to calculate $\hat{\mathbf{x}}^+$ and $\mathbf{P}^+$ at the end of each time step.

It should be noted here that the computational cost of performing a matrix logarithm and matrix exponential in (13) and (18) causes the runtime for the BFF to be nearly double that of the GFF as the algorithms are implemented for this work. With additional care, this difference can be mitigated.

The following sections detail simulated examples to demonstrate the BFF and compare it against the GFF.

V. Linear System

This section covers the results from running both the BFF and GFF on a linear system of equations over a 50 step time period with a measurement every step. Each filter is run using the same truth data (i.e. true state and measurements) for 100 Monte Carlo (MC) runs with a varying numbers of particles. The number of pseudotime steps used is $N_\lambda = 2$.

Consider the following system of equations,

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{F}\mathbf{x}_k + \boldsymbol{\nu}, \quad \mathbf{F} = \begin{bmatrix} 0 & 0.1 \\ -1 & 0 \end{bmatrix}, \\ \mathbf{y}_k &= \mathbf{H}\mathbf{x}_k + \boldsymbol{\eta}, \quad \mathbf{H} = \begin{bmatrix} \frac{1}{2} & 0 \end{bmatrix}, \end{aligned} \tag{22}$$

that contains the governing equations for both the true system of equations as well as the system of equations in the filter. The initials state is drawn from $\mathbf{x}_0 \sim \mathcal{N}([1,-1]^T, \boldsymbol{I})$, with errors $\boldsymbol{\nu} \sim \mathcal{N}(\mathbf{0}, 0.01\boldsymbol{I})$, and $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.

The Root Mean Square Error (RMSE) is used to measure the error in the estimate from the filters. RMSE at each time step is defined as:

$$\mathrm{RMSE}_k = \sqrt{\frac{1}{N_{\mathrm{MC}}}\sum_{i=1}^{N_{\mathrm{MC}}}(\hat{\mathbf{x}}_k^i - \mathbf{x}_k^i)^2}, \tag{23}$$

where N_{MC} is the number of MC runs.

The Scaled Normalized Estimation Error Squared (SNEES) is used to measure the statistical consistency of the state estimates and covariance values from the filters. The SNEES at each time step is defined as:

$$\mathrm{SNEES}_k = \frac{1}{nN_{\mathrm{MC}}}\sum_{i=1}^{N_{\mathrm{MC}}}(\hat{\mathbf{x}}_k^i - \mathbf{x}_k^i)^T(\mathbf{P}_k^i)^{-1}(\sim). \tag{24}$$

It should be noted that a RMSE of 0 implies no error and a SNEES value of 1 indicates perfect consistency. To simplify the results both the RMSE and SNEES are taken and averaged over the time span of the simulation.

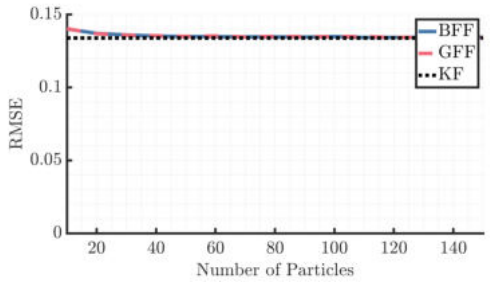

Fig. 2. Average RMSE of the filter estimate compared with the true state across simulation time and across MC runs versus number of particles used in the filter for the BFF and GFF. The black line shows a similar RMSE value from 100 MC runs through a KF.

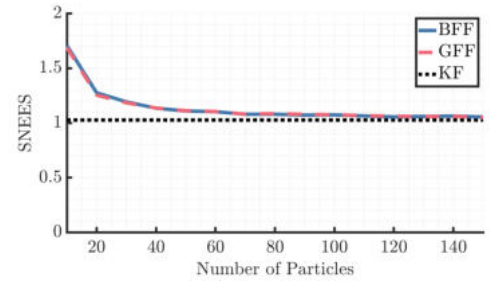

Fig. 3. Average SNEES of the filter estimate compared with the true state across simulation time and across MC runs versus number of particles used in the filter for the BFF and GFF. The black line shows a similar SNEES value from 100 MC runs through a KF.

Figures 2 and 3 show the results from running a 100 run MC simulation using the system of equations in (22). When varying the number of particles in both filters, the RMSE and SNEES both improve with a classic elbow in the curve where the benefit of adding more particles becomes lessened. In both figures, this occurs around 40 particles for both the BFF and GFF. As the number of particles in the filters increases, the solution that the filter yields approaches the solution from the Kalman filter, both in the RMSE and in the SNEES. It should be noted again that the BFF and GFF algorithms utilize an ensemble covariance rather than relying on a companion filter to obtain their estimates of the covariance of the state. Utilizing an ensemble covariance allows the filters to evolve their covariance estimates based directly on the evolution of the particles themselves, but can degrade the performance on a linear system for small numbers of particles.

Now briefly examining the stiffness of the two filters, Table I shows the norm of the initial step size taken by the BFF and GFF starting from the mean of the prior distribution at the initial time step. It is important to note that the substep size is the same for both filters, $d\lambda = \frac{1}{\lambda}$. The BFF shows a more moderate initial update, indicating that the BFF is less stiff than the GFF in this example.

It is clear that the BFF yields very similar results to the GFF when applied to linear systems. In the next section, the results for a nonlinear system are shown.

TABLE I
INITIAL SUBSTEP UPDATE STARTING FROM THE MEAN OF THE INITIAL DISTRIBUTION FOR THE BFF AND GFF FOR THE LINEAR SYSTEM.

Filter	$\|\|d\mathbf{x}_1\|\|$
BFF	**0.2340**
GFF	0.4985

VI. Two Dimensional Range Observation System

This example displays the results from applying the BFF and GFF to a nonlinear system. Similar to the relationship between the KF and the EKF, the derivation of the equations that govern the BFF for a nonlinear system is nearly the same as for a linear system. In the derivation in Section III the measurement is a linear function of the state. Now it is assumed that the measurement model can be approximated by a first-order Taylor expansion about the most recent estimate. While this is a similar assumption for the BFF as the EKF, the BFF takes small pseudotime steps towards the posterior distribution, which allows for the recursive recalculation of $\mathbf{H}$ as the state moves through the update steps. The number of pseudotime steps used in this scenario is $N_\lambda = 10$. All of the same equations from Section III still apply with the caveat that $\mathbf{H}$ is now a function of $\hat{\mathbf{x}}$.

Consider a range measurement for a two dimensional state space:

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad y = \sqrt{x_1^2 + x_2^2} + \eta, \\ \mathbf{H} &= \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}} = \frac{1}{\|\hat{\mathbf{x}}\|} \begin{bmatrix} \hat{x}_1 & \hat{x}_2 \end{bmatrix}, \end{aligned} \tag{25}$$

In this example, the particles are taken through a single measurement update without time propagation. The initial distribution (prior distribution) of the particles in this problem is Gaussian, centered on $[-3, 0]^T$ with an uncertainty of $\sigma_1^2 = \sigma_2^2 = 1$, $\sigma_1\sigma_2 = \sigma_2\sigma_1 = 0.5$. The measurement uncertainty for this system is $R = 0.01$. By Bayes Rule, the posterior distribution is obtained by multiplying the prior distribution by the measurement likelihood distribution:

$$\begin{aligned} p_{\mathbf{x}} &= \frac{1}{2\pi\sqrt{|P_0|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}})^T P_0^{-1}(\mathbf{x}-\bar{\mathbf{x}})\right) \\ &= \mathcal{N}\left(\begin{bmatrix} -3 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}\right), \\ p_{y|\mathbf{x}} &= \frac{1}{2\pi\sqrt{R}} \exp\left(-\frac{1}{2R}\left(y - \sqrt{x_1^2 + x_2^2}\right)^2\right), \\ p_{\mathbf{x},y} &= \frac{1}{c} p_{\mathbf{x}} p_{y|\mathbf{x}}, \end{aligned} \tag{26}$$

where $p_{\mathbf{x}}$ is the initial bivariate Gaussian pdf, $p_{y|\mathbf{x}}$ is the measurement likelihood pdf given the state, $p_{\mathbf{x},y}$ is the posterior pdf of the state, and c is a normalization constant to ensure $p_{\mathbf{x},y}$ is a valid pdf.

Figures 4 and 5 show contour plots of the prior, measurement likelihood, and posterior distributions for this example.

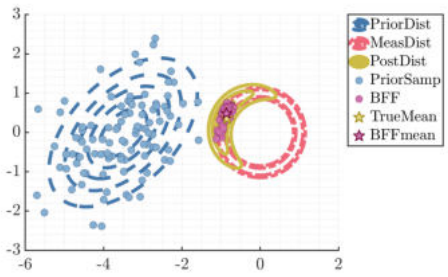

Fig. 4. BFF results for the range observation example. The blue contour lines and circles show the prior distribution and samples. The red contour lines show the measurement distribution. The yellow contour lines and star show the true posterior distribution and its mean. The purple circles and star correspond to the results from running the initial samples through the measurement update of the BFF and their mean.

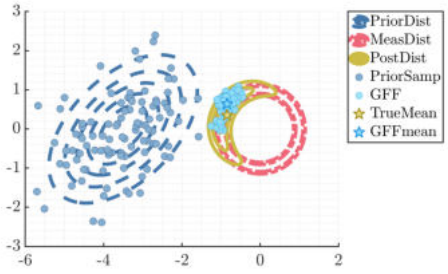

Fig. 5. GFF results for range observation example. The cyan circles and star correspond to the results from running the initial samples through the measurement update of the GFF and their mean.

It should be noted that the true posterior distribution is not analytically available. The contour plots are generated by computing the values of $p_{\mathbf{x}}$, $p_{y|\mathbf{x}}$, and $p_{\mathbf{x},y}$ on a fine grid using (26). Figure 4 shows the results of the BFF update, and Figure 5 shows the results of the GFF update. Inspecting Figures 4 and 5, it is clear that both the BFF and GFF perform well in representing the posterior distribution.

To quantify how well the BFF and GFF represent the posterior distribution, a discrete Kullback-Leibler divergence (KL-divergence) test is conducted. Discrete samples of the true posterior distribution are generated on a fine grid over the space in Figures 4 and 5. The normalization constant is calculated by numerically integrating the product of the prior and measurement likelihood distributions over the grid. The particles from the particle flow filters act as their own equally-weighted discrete samples. The KL-divergence determines the divergence between the two distributions. A KL-divergence value of 0 indicates that the distributions are equal almost everywhere [25]. The results can be found in Table II.

The KL-divergence of the BFF compared with the true posterior distribution is lower than the KL-divergence of the GFF. This is a clear indication that the BFF produces results that are more consistent with the posterior distribution. With this result in mind, Figures 4 and 5 also appear to support that the results from the BFF more closely resemble the posterior distribution. The cyan circles that represent the GFF spill over

TABLE II
RESULTS FROM A KL-DIVERGENCE TEST USING 1000 POINTS FROM THE BFF AND GFF RESPECTIVELY.

Filter	KL-divergence
BFF	**0.3266**
GFF	0.4720

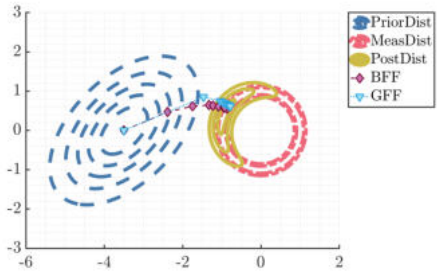

Fig. 6. Plot to visualize the progression of state estimates from the BFF and GFF in a single measurement update with 10 substeps of size $\frac{1}{10}$.

the edges of the yellow posterior contour while the purple circles of the BFF are more well contained.

One possible explanation as to why the BFF yields more statistically consistent results is due to the stiffness of the two filters. A diagram showing the substeps of the BFF and GFF in a single measurement update from the mean is shown in Figure 6. Additionally, Table III shows the magnitude of the initial update applied by each filter.

It is well know that particle flows can be extremely stiff [16]. This can be observed by looking at the size of the update during the first substep for both the BFF and GFF. While the BFF's initial substep update is still much larger than its subsequent substep updates, the GFF's update is nearly twice as large as the BFF's. Additionally, the BFF has an overall smoother flow towards the desired posterior distribution. Again, it is important to note that the substep size is the same for both filters, $d\lambda = \frac{1}{\lambda}$. This behavior indicates that the BFF is less stiff than the GFF in this scenario.

The next section covers the results from a nonlinear system with both nonlinear dynamics and nonlinear measurements.

TABLE III
INITIAL SUBSTEP UPDATE STARTING FROM THE MEAN OF THE INITIAL DISTRIBUTION FOR THE BFF AND GFF FOR THE 2D RANGE OBSERVATION SYSTEM.

Filter	$\|\|d\mathbf{x}_1\|\|$
BFF	**1.207**
GFF	2.206

VII. Ikeda Map

Consider the following system of equations:

$$\mathbf{x}_{k+1} = C_2 \begin{bmatrix} \cos(z_k) & -\sin(z_k) \\ \sin(z_k) & \cos(z_k) \end{bmatrix} \mathbf{x}_k + \begin{bmatrix} C_4 \\ 0 \end{bmatrix} + \boldsymbol{\nu},$$
$$z_k = C_1 - \frac{C_3}{1 + \|\mathbf{x}_k\|^2}, \tag{27}$$
$$y = \sqrt{x_1^2 + x_2^2} + \eta, \quad \mathbf{H} = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_k} = \frac{1}{\|\hat{\mathbf{x}}\|} \begin{bmatrix} \hat{x}_1 & \hat{x}_2 \end{bmatrix},$$

where $C_1 = 0.4$, $C_2 = C_4 = 0.9$, and $C_3 = 6$. In this example, the initial state is drawn from $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, with errors $\boldsymbol{\nu} \sim \mathcal{N}(\mathbf{0}, 0.1\mathbf{I})$, and $\boldsymbol{\eta} \sim \mathcal{N}(0, 1)$.

This system in (27) is used to generate both the truth data and in the filters themselves. A simulation with 100 MC runs is conducted and the results are found in Figures 7 and 8.

In Figure 7, in addition to the results for the BFF and GFF filters, the results from a regularized particle filter (RPF) with 10,000 particles are also included. This result acts as a lower bound for what the RMSE (23) might be for the system. In Figure 7, the BFF and GFF display similar performance in the accuracy of their estimates.

This section covers the results from testing the BFF and GFF on a nonlinear system with dynamics defined by a 2D Ikeda Map [26] and range measurements over a 50 step time period with measurements every step. The number of pseudotime steps used in this scenario is $N_\lambda = 10$. This system is extremely nonlinear and thus is an excellent test for the BFF.

In Figure 8, it shows that the BFF provides more statistically consistent results than the GFF as the number of particles considered in the filter is increased. At around 30 particles, the GFF transitions from statistically under representing the amount of noise in the system to over representing the amount of noise in the system. The BFF on the other hand, achieves a closer asymptote to the ideal SNEES (24) value of 1. This indicates that the flow solution in the BFF algorithm is more statistically consistent than that of the GFF.

Finally, briefly examining the stiffness of the BFF and GFF by examining the first substep of the measurement update from the initial prior distribution shown in Table IV. The first substep update provided by the GFF is nearly twice as large as the first substep update provided by the BFF. This is an indication that the BFF is less stiff than the GFF.

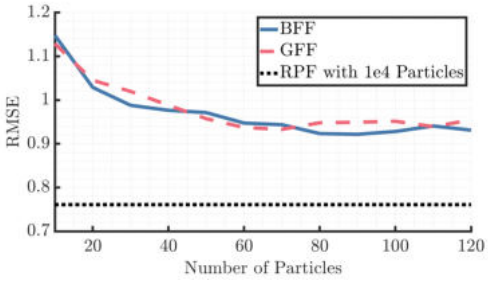

Fig. 7. Average RMSE of the filter estimate compared with the true state across simulation time and across MC runs versus number of particles used in the filter for the BFF and GFF. The black line shows a similar RMSE value from 100 MC runs through a RPF with 10,000 particles.

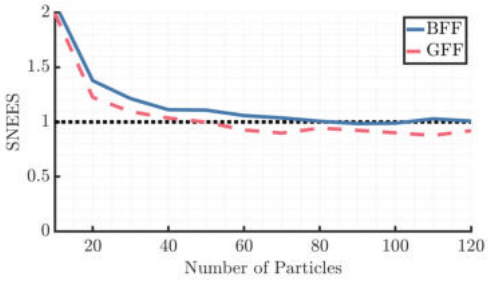

Fig. 8. Average SNEES of the filter estimate compared with the true state across simulation time and across MC runs versus number of particles used in the filter for the BFF and GFF. The black line shows a SNEES value of 1.

TABLE IV
Initial substep update starting from the mean of the initial distribution for the BFF and GFF for the Ikeda Map system.

Filter	$\|\|d\mathbf{x}_1\|\|$
BFF	**2.154**
GFF	4.568

VIII. Conclusions

The BFF is a particle flow filter that leverages the Kalman update equations to transport particles from the prior distribution to the posterior distribution for a given measurement. The BFF is compared against the GFF in three examples. It is shown that the BFF is capable of providing more statistically consistent results than the GFF. These results give confidence that the BFF is a good choice for highly nonlinear systems.

The results presented in Section V demonstrate the BFF provides an estimate for a linear, Gaussian system just as well as the GFF. Section VI demonstrates that the BFF provides a more statistically consistent estimate than the GFF for a system that transforms from a Gaussian to a non-Gaussian due to a nonlinear measurement. Section VII shows that the BFF is capable of providing a more statistically consistent estimate of a system with both nonlinear dynamics and nonlinear measurements.

In each scenario it is clear that the SDE that governs the BFF is less stiff than the GFF. In each scenario, the initial substep update of the GFF is nearly twice the size of the BFF's substep update. This analysis demonstrates that the BFF should be studied further in order to understand this persistent difference in performance. However, the BFF's initial substep update is still large compared to the subsequent substep updates. In future work, analysis should be done to examine how the BFF might benefit from a variable substep size.

Another avenue for future work, is to examine other ways to to reduce the computational complexity of the BFF. The current form of the BFF calls for a matrix log and a matrix exponential at each update step. It is possible that the need

for at least one of these operations may be obviated by minor changes to the derivation in Section III.

Not only is the derivation of the BFF accessible, but, more importantly, the results it produces are competitive. More studies should be conducted to better understand the conditions that cause the BFF to yield more consistent results than the GFF. Additionally, more studies should be performed comparing the BFF against additional nonlinear filters.

REFERENCES

[1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME, Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[2] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *Transactions of the ASME, Journal of Basic Engineering*, vol. 83, no. 1, pp. 95–108, 1961.

[3] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software*. New York, NY, USA: Wiley, 2004.

[4] B. D. Tapley, B. Schutz, and B. G. H., *Statistical Orbit Determination*. Burlington, MA, USA: Elsevier Academic Press, 2004.

[5] A. Doucet, N. De Freitas, N. Gordon, and et al, *Sequential Monte Carlo methods in practice*. Springer, 2001, vol. 1.

[6] 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, 107–113(6), 2 Apr. 1993, ISSN: 0956-375X.

[7] F. Daum and J. Huang, "Nonlinear filters with log-homotopy," *SPIE: Signal and Data Processing of Small Targets*, vol. 6699, 2007.

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

[9] T. Ding and M. Coates, "Implementation of the daum-huang exact-flow particle filter," *2012 IEEE Statistical Signal Processing Workshop (SSP)*, pp. 257–260, 2012.

[10] F. Daum, J. Huang, and A. Noushin, "Coulomb's law particle flow for nonlinear filters," *Signal and Data Processing of Small Targets 2011*, pp. 99–108, 2011.

[11] F. Daum and J. Huang, "Generalized particle flow for nonlinear filters," *Signal and Data Processing of Small Targets*, vol. 7698, 2010.

[12] S. Pal and M. Coates, "Particle flow particle filter using gromov's method," *2019 IEEE 8th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pp. 634–638, 2019.

[13] F. Daum and J. Huang, "Particle flow for nonlinear filters, bayesian decisions and transport," *16th International Conference on Information Fusion*, pp. 1072–1079, 2013.

[14] F. Daum and J. Huang, "Particle flow for nonlinear filters, bayesian decisions and transport," *Signal Processing, Sensor Fusion, and Target Recognition XXII*, vol. 8745, 2013.

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

[16] L. Dai and F. Daum, "On the design of stochastic particle flow filters," *IEEE Transactions on Aerospace and Electronic Systems*, 2022.

[17] P. S. Maybeck, *Stochastic models, estimation, and control, Volume 1*. New York, NY, USA: Academic Press, Inc., 1979.

[18] 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.

[19] 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.

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

[21] O. Cappé, E. Moulines, and T. Ryden, *Inference in hidden markov models*. Newyork, NY, USA: Springer, 2007.

[22] F. Daum, J. Huang, and A. Noushin, "Gromov's method for bayesian stochastic particle flow: A simple exact formula for q," *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pp. 540–545, 2016.

[23] F. Daum, J. Huang, and A. Noushin, "New theory and numerical results for gromov's method for stochastic particle flow filters," in *2018 21st International Conference on Information Fusion (FUSION)*, IEEE, 2018, pp. 108–115.

[24] K. Michaelson, A. A. Popov, and R. Zanetti, "Bayesian recursive update for ensemble kalman filters," *arXiv preprint arXiv:2310.18442*, 2023.

[25] C. Archambeau, D. Cornford, M. Opper, and J. Shawe-Taylor, "Gaussian process approximations of stochastic differential equations," in *Gaussian Processes in Practice*, N. D. Lawrence, A. Schwaighofer, and J. Quiñonero Candela, Eds., ser. Proceedings of Machine Learning Research, vol. 1, Bletchley Park, UK: PMLR, Dec. 2007, pp. 1–16. [Online]. Available: https://proceedings.mlr.press/v1/archambeau07a.html.

[26] 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, ISSN: 0030-4018. DOI: https://doi.org/10.1016/0030-4018(79)90090-7. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0030401879900907.