Particle Flow with a Continuous Formulation of the Nonlinear Measurement Update

Kristen Michaelson*, Andrey A. Popov†, Renato Zanetti*†, Kyle J. DeMars‡
*Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, Texas 78712–1221
Email: kmichaelson@utexas.edu, renato@utexas.edu
†Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, Texas 78712–1221
Email: andrey.a.popov@utexas.edu
‡Department of Aerospace Engineering, Texas A&M University, College Station, TX 77843–3141
Email: demars@tamu.edu

***Abstract*—The incorporation of nonlinear measurement information plays an important role in Bayesian state estimation for real-word systems. While many methods exist for propagating states through continuous-time nonlinear dynamics, a complementary continuous solution for discrete-time nonlinear measurements has so far remained elusive. Building on intuition from our previous work, the Bayesian Recursive Update Filter, we formulate the nonlinear measurement update as an ordinary differential equation (ODE). This formulation naturally extends to particle flow. We define two particle flows: the first is a deterministic flow based on the ODE solution, and the second is stochastic; the numerical integration contains a diffusion term. The proposed particle flows demonstrate excellent performance on a system with deterministic dynamics and a highly accurate nonlinear measurement, a setting known to be challenging for particle filters.**

***Index Terms*—statistical estimation, nonlinear filtering, particle flow, bayesian recursive update**

I. Introduction

Designing tractable filtering algorithms for nonlinear systems is a perennial challenge in statistical estimation. Uncertainty distributions are rarely available in closed form, leading to sub-optimal state estimates. While linearization and Gaussian approximations may be used to guess the correct solution, there is no optimality guarantee.

One existing class of nonlinear filters is the particle filters [1]. Particle filters represent the uncertainty distribution with a set of state estimates, the *particles*, which are taken to be samples of the uncertainty distribution. The particles are propagated through the dynamics model, forming the prior distribution. When a measurement arrives, they are moved and/or re-weighted based on the measurement likelihood. Often, a resampling step is employed, which replicates particles with higher weight and removes particles with lower weight.

Ideally, no weighting or resampling would be required. At measurement time, the particles would simply move from the propagated prior distribution to the Bayesian posterior distribution. Then, again, to the next prior, and the next posterior, and so on, as the dynamics evolve and measurements are received. This is the objective of *particle flow*. The term "particle flow" traditionally refers to the measurement update; the idea of steadily moving (i.e., "flowing") particles from the prior distribution to the posterior distribution given the measurement information.

Daum-Huang Exact flow was an early particle flow [2], [3]. Later, Gromov flow was introduced, which includes a diffusion term [4]. These flows are known as log-homotopy flows, since they approximate a homotopy between the prior distribution and the posterior distribution. Numerous log-homotopy flows have been developed [5]. More recently, particle flows based on Stein variational gradient descent [6] and projected cumulative distributions [7] have also been proposed.

In this work, we expand on previous work by using the Bayesian Recursive Update Filter (BRUF) [8] to induce a particle flow. The BRUF relaxes the Extended Kalman Filter (EKF) update by applying a series of updates, each with inflated measurement noise covariance. This idea is related to likelihood tempering [9], the damped iterated EKF update proposed in [10], and progressive Gaussian filtering [11]. Perhaps the most similar existing algorithm is multiple data assimilation (MDA) in ensemble filtering [12], though the BRUF was derived independently.

While particle flow is a natural extension of the BRUF—and, indeed, we have already developed an ensemble Kalman filter (EnKF) version [13], [14]—we revisit the derivation of the BRUF, expanding it to arrive at an ODE formulation. We then introduce two particle flows based on the ODE solution. The first is deterministic, and the second is stochastic. Deriving a stochastic flow was of particular interest for comparison to existing stochastic flows.

The remainder of this article is organized as follows: Section II presents the derivation of the ODE formulation of the nonlinear measurement update; Section III defines the two particle flows and shows results for a Gaussian-distributed prior state estimate with a range measurement; Section IV presents a dynamics example; and Section V presents conclusions and future work.

This material is based on research sponsored by the Air Force Office of Scientific Research (AFOSR) under agreement number FA9550-23-1-0646, *Create the Future Independent Research Effort (CFIRE).*

II. Background

Consider nonlinear measurement model

$$y = h(x) + \nu \tag{1}$$

where

$$\nu \sim \mathcal{N}(0, R) \tag{2}$$

is zero-mean additive Gaussian noise. Assume a state estimate m is available with prior distribution

$$x \sim \mathcal{N}(m, P) \tag{3}$$

where P is the covariance of our uncertainty.

The Extended Kalman Filter (EKF) update is

$$\begin{aligned} \hat{x} &= m + K(y - h(m)) \\ \hat{P} &= (I - K\tilde{H})P \end{aligned} \tag{4}$$

where K is the Kalman gain:

$$K = P\tilde{H}^T(\tilde{H}P\tilde{H}^T + R)^{-1}. \tag{5}$$

The Kalman gain and the covariance update depend on the measurement Jacobian,

$$\tilde{H} = \left.\frac{\mathrm{d}h(x)}{\mathrm{d}x}\right|_{x=m}. \tag{6}$$

If the linearization (6) is a poor approximation of the measurement function, the EKF may diverge. Further, the EKF update provides a Gaussian approximation of the Bayesian posterior distribution. It cannot capture higher-order moments.

We would like to define an alternative recursive measurement update in which the state estimate moves slowly from m to $\hat{x}$. We will use this iterative scheme to induce a particle flow. The updated particle distribution will better match the true statistics of the Bayesian posterior distribution than a single state estimate with a Gaussian uncertainty approximation.

Beginning with the full expression for the Bayesian posterior distribution,

$$\begin{aligned} p(x|y) = C \cdot \exp\Big\{ -\frac{1}{2}(m-x)^T P^{-1}(m-x)\Big\} \\ \cdot \exp\Big\{ -\frac{1}{2}(y-h(x))^T R^{-1}(y-h(x))\Big\} \end{aligned} \tag{7}$$

we start by re-expressing the argument of the measurement likelihood function as a weighted sum:

$$\begin{aligned} p(x|y) = C \cdot \exp\Big\{ -\frac{1}{2}(m-x)^T P^{-1}(m-x)\Big\} \\ \cdot \exp\Big\{ -\frac{1}{2}\sum_{i=0}^{n-1} \alpha_i (y-h(x))^T R^{-1}(y-h(x))\Big\} \end{aligned} \tag{8}$$

where

$$\sum_{i=0}^{n-1} \alpha_i = 1. \tag{9}$$

Using a convenient exponential identity, we can now express the posterior distribution as the product of the prior distribution and n likelihood functions.

$$\begin{aligned} p(x|y) = C \cdot \exp\Big\{ -\frac{1}{2}(m-x)^T P^{-1}(m-x)\Big\} \\ \cdot \prod_{i=0}^{n-1} \exp\Big\{ -\frac{1}{2}\alpha_i (y-h(x))^T R^{-1}(y-h(x))\Big\} \end{aligned} \tag{10}$$

Alternatively, we can write:

$$\begin{aligned} p(x|y) = C \cdot \exp\Big\{ -\frac{1}{2}(m-x)^T P^{-1}(m-x)\Big\} \\ \cdot \prod_{i=0}^{n-1} \exp\Big\{ -\frac{1}{2}(y-h(x))^T \left(\frac{1}{\alpha_i}R\right)^{-1} (y-h(x))\Big\}. \end{aligned} \tag{11}$$

Since each $\alpha_i \ll 1$, the term $\frac{1}{\alpha_i}R$ is an *inflation* of the measurement noise covariance. For example, if $\alpha_i = \frac{1}{n}$ for all i, then the inflated measurement noise covariance matrix in each likelihood function is nR.

We would like to approximate the product of Gaussians in (11) as a single Gaussian with mean $\hat{x}$ and covariance $\hat{P}$. Beginning with the prior and the first likelihood function, we can write

$$\begin{aligned} p(x|y) \approx C \cdot \exp\Big\{ -\frac{1}{2}(\hat{x}^{(1)}-x)^T P_1^{-1}(\hat{x}^{(1)}-x)\Big\} \\ \cdot \prod_{i=1}^{n-1} \exp\Big\{ -\frac{1}{2}(y-h(x))^T \left(\frac{1}{\alpha_i}R\right)^{-1} (y-h(x))\Big\} \end{aligned} \tag{12}$$

where the first exponential function now has the mean and covariance values

$$\hat{x}^{(1)} = m + \tilde{K}_0(y - h(m)) \tag{13}$$

$$P_1 = (I - \tilde{K}_0\tilde{H}_0)P \tag{14}$$

$$\tilde{K}_0 = P\tilde{H}_0^T\left(\tilde{H}_0 P\tilde{H}_0^T + \frac{1}{\alpha_0}R\right)^{-1} \tag{15}$$

and $\tilde{H}_0$ is evaluated at $x = m$. The appropriate normalization constants have been absorbed into C.

This approximation holds as long as the state update is small enough that $h(\hat{x}^{(1)}) \approx h(m) + \tilde{H}_0\left(\hat{x}^{(1)} - m\right)$; in other words, as long as the updated state stays within the linear region of $h(x)$ evaluated at $x = m$. If the linear approximation is valid, then x and y are jointly Gaussian, and $\hat{x}^{(1)}$ is the minimum mean-square-error (MMSE) and maximum *a posteriori* (MAP) estimate of x given inflated measurement noise covariance $\frac{1}{\alpha_0}R$ [15] (Chs. 2–3). The linearization is certainly valid if $\frac{1}{\alpha_0}R$ is very large. The measurement y will hardly be trusted, and the estimate $\hat{x}^{(1)}$ will remain close to the prior mean. However, by making a Gaussian approximation of this first posterior distribution, we lose higher-order information about the true uncertainty distribution of the state.

Continuing, we can recursively define:

$$\hat{x}^{(i+1)} = \hat{x}^{(i)} + \tilde{K}_i\left(y - h\left(\hat{x}^{(i)}\right)\right) \tag{16}$$

$$P_{i+1} = (I - \tilde{K}_i \tilde{H}_i) P_i \tag{17}$$

$$\tilde{K}_i = P_i \tilde{H}_i^T \left(\tilde{H}_i P_i \tilde{H}_i^T + \frac{1}{\alpha_i} R \right)^{-1} \tag{18}$$

with $\hat{x}^{(0)} = m$ and $P_0 = P$. This leads to the Gaussian approximation of the posterior distribution:

$$p(x|y) \approx C \cdot \exp\Big\{ -\frac{1}{2} (\hat{x}^{(n)} - x)^T P_n^{-1} (\hat{x}^{(n)} - x) \Big\}. \tag{19}$$

We have now collapsed the right-hand side of (11) into a single Gaussian, fully incorporating the measurement information.

We now face a classic tradeoff between step size α_i and computational cost. Clearly, the linear region criteria discussed above hold for $\alpha_i \to 0$. In this case, $\frac{1}{\alpha_i} R \to \infty$, and the state update goes to zero. We would like to automatically choose step sizes α_i small enough that the linear approximation holds, but large enough that we can solve the problem in a reasonable number of steps. This is difficult to do for the recursion in (16)-(18). The step α_i appears in the denominator in (18); the term $\frac{1}{\alpha_i} R$ is then added to another matrix, and the whole expression is inverted again. Further, the expression for $\hat{x}^{(i+1)}$ contains a nonlinear function of $\hat{x}^{(i)}$.

As a more palatable alternative to the Kalman update in (16)-(18), consider the forward form of the Kalman gain

$$\tilde{K}_i = P_{i+1} \tilde{H}_i^T R^{-1} \tag{20}$$

where the gain is now expressed in terms of the *updated* covariance matrix. We can now express the update in (16)-(18) as

$$\hat{x}^{(i+1)} = \hat{x}^{(i)} + \alpha_i P_{i+1} \tilde{H}_i^T R^{-1} \left(y - h\left(\hat{x}^{(i)} \right) \right) \tag{21}$$

$$P_{i+1} = P_i - \alpha_i P_{i+1} \tilde{H}_i^T R^{-1} \tilde{H}_i P_i \tag{22}$$

where we have replaced the Kalman gain $\tilde{K}_i$ with the form in (20), and the terms α_i have been pulled out of the measurement noise covariance inversion.

We can now write down differential equations for $\hat{x}$ and P by evaluating

$$\begin{aligned} \frac{\mathrm{d}\hat{x}^{(i)}}{\mathrm{d}\alpha} &= \lim_{\alpha_i \to 0} \frac{\hat{x}^{(i+1)} - \hat{x}^{(i)}}{\alpha_i} \\ &= \lim_{\alpha_i \to 0} \frac{\alpha_i P_{i+1} \tilde{H}_i^T R^{-1} \left(y - h\left(\hat{x}^{(i)} \right) \right)}{\alpha_i} \\ &= \lim_{\alpha_i \to 0} P_{i+1} \tilde{H}_i^T R^{-1} \left(y - h\left(\hat{x}^{(i)} \right) \right) \end{aligned} \tag{23}$$

$$\begin{aligned} \frac{\mathrm{d}P_i}{\mathrm{d}\alpha} &= \lim_{\alpha_i \to 0} \frac{P_{i+1} - P_i}{\alpha_i} \\ &= \lim_{\alpha_i \to 0} \frac{-\alpha_i P_{i+1} \tilde{H}_i^T R^{-1} \tilde{H}_i P_i}{\alpha_i} \\ &= \lim_{\alpha_i \to 0} -P_{i+1} \tilde{H}_i^T R^{-1} \tilde{H}_i P_i \end{aligned} \tag{24}$$

The key is to recognize that as $\alpha_i \to 0$, then $P_{i+1} \to P_i$. Hence, removing time indices and changing the discrete step α_i to the pseudotime parameter τ,

$$\frac{\mathrm{d}x}{\mathrm{d}\tau} = P \tilde{H}^T R^{-1} \left(y - h(x) \right) \tag{25}$$

$$\frac{\mathrm{d}P}{\mathrm{d}\tau} = -P \tilde{H}^T R^{-1} \tilde{H} P \tag{26}$$

where

$$\tilde{H} = \frac{\mathrm{d}h(x)}{\mathrm{d}x}. \tag{27}$$

The measurement update is computed by numerically integrating (25)-(26) on the interval $\tau \in [0, 1]$.

The terms on the right-hand side of (25)-(26) are analogous to the measurement update terms in the Kalman-Bucy filter. The Kalman-Bucy filter is a continuous-time filter. The state dynamics evolve in time as measurements are received in a constant stream. The use of limit evaluations to derive the differential equations (25)-(26) follows the derivation of the Kalman-Bucy filter in [15] (Ch. 9). In the Appendix, we show that (25)-(26) can also be derived from the information filter equations.

It is important to note that Eqs. (25)-(26) form a system of differential equations that must be integrated together. They are inextricably coupled, since the state evolution depends on the covariance, and both the state and the covariance dynamics depend on the current state to evaluate $\tilde{H}$.

All numerical ODE solutions presented in this work were computed in MATLAB using `ode45` with the default settings [16], [17]. Sections III-IV also contain results generated by a numerical stochastic differential equation (SDE) solver, Rejection Sampling with Memory (RSwM1) [18]. The absolute and relative tolerances used by RSwM1 were chosen to match the `ode45` defaults.

III. Particle Flow

We begin with a motivating example for generalizing (25)-(26) to particle flow. While integrating (25)-(26) for a single state estimate gives a Gaussian approximation of the posterior distribution, we will show that applying (25)-(26) to a set of particles with a Gaussian prior distribution can successfully redistribute the particles over a non-Gaussian posterior.

Consider the range measurement

$$y = \|x\| + \nu \tag{28}$$

where $\nu \sim \mathcal{N}(0, 0.1^2)$. The measurement Jacobian is

$$\tilde{H} = \frac{x^T}{\|x\|}. \tag{29}$$

Fig. 1 shows the state trajectory that results from numerically integrating (25)-(26) given measurement

$$y = 1 \tag{30}$$

with prior state estimate

$$x = \begin{bmatrix} -3.5 & 0 \end{bmatrix}^T \tag{31}$$

and covariance

$$P = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}. \tag{32}$$

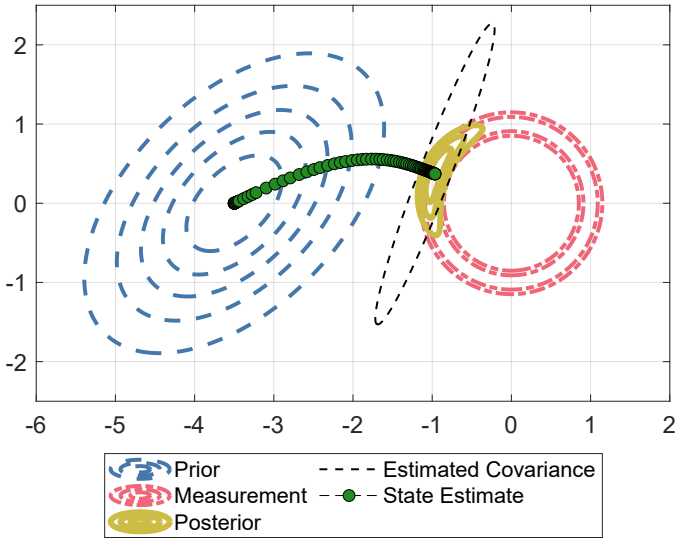


Fig. 1: Evolution of state estimate for range measurement example. The final 2σ-covariance ellipse is show in black. The measurement likelihood distribution is a ring centered at the origin. This produces a crescent-shaped *a posteriori* distribution.

Beginning at the left of Fig. 1, the state estimate moves smoothly from the prior distribution to the posterior distribution. The final covariance matrix matches the posterior distribution along the direction of the range vector. The estimate is underconfident in the cross-range direction. We have shown in previous work that these state and covariance values closely match the result of the IEKF update for this example [14]. The IEKF cannot solve this problem without line search [14], [19].

The following sections present two alternative particle flow formulations based on (25)-(26). The first is a deterministic flow that takes advantage of a diffusion technique from ensemble filtering. The second is a stochastic flow that includes a random diffusion term at each integration step.

A. Particle flow based on ODE solution

Naively, one may define a flow by using an ODE solver to integrate (25)-(26) for each particle. However, this is prohibitively expensive for filters with large numbers of particles, and especially for systems with large state spaces. Instead, we opt to solve the ODE beginning from the prior mean and the ensemble covariance. Saving only the series of time steps from this initial solve, we then execute a full BRUF update for each particle.

Given the set of particles

$$X = \{x_1, \ldots, x_{n_p}\}, \tag{33}$$

the steps are as follows:

1) Compute the prior mean, m, and the sample covariance

$$P = \frac{1}{n_p - 1} \sum_{i=1}^{n_p} (x_i - m)(x_i - m)^T \tag{34}$$

of the particles.

2) Solve Eqs. (25)-(26) with initial conditions m and P. Obtain time series $\left\{\Delta\tau^{(1)} \ \ldots \ \Delta\tau^{(n_t)}\right\}$, where

$$\sum_{k=1}^{n_\tau} \Delta\tau^{(k)} = 1. \tag{35}$$

3) Generate a set of perturbed measurements

$$\tilde{y}_i = y + \eta \tag{36}$$

where $\eta \sim \mathcal{N}(0, R)$.

4) Complete a full BRUF update for each particle using the time series from Step 2:

$$x_i^{(k+1)} = x_i^{(k)} + K_i^{(k)} \left(\tilde{y}_i - h\left(x_i^{(k)}\right)\right) \tag{37}$$

$$P_i^{(k+1)} = \left(I - K_i^{(k)} \tilde{H}_i^{(k)}\right) P^{(k)} \tag{38}$$

where

$$K_i^{(k)} = P_i^{(k)} \tilde{H}_i^{(k)T} \left(\tilde{H}_i^{(k)} P_i^{(k)} \tilde{H}_i^{(k)T} + \tilde{R}^{(k)}\right)^{-1} \tag{39}$$

$$\tilde{R}^{(k)} = \frac{1}{\Delta\tau^{(k)}} R \tag{40}$$

Fig. 2 shows the result of this process for a set of 100 particles drawn from the prior distribution in Fig. 1. The initial set of particles is shown in black. The dashed lines trace their trajectories over the course of the update. The updated particles are shown in light green, nicely settled on the posterior distribution.

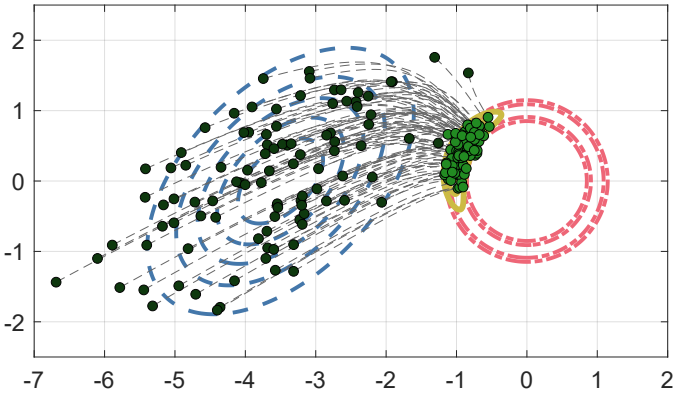


Fig. 2: Particle flow induced by ODE solution

Step 3 is borrowed from ensemble filtering [13], [20], [21]. Fig. 3 shows the result of the algorithm above for $\tilde{y}_i = y$. The initial particle locations are the same as in Fig. 2. If the same measurement value $y = 1$ is used for each particle, then the particles settle on the circle $\|x\| = 1$. This maximizes the measurement likelihood for each individual particle, but it does not lead to a healthy distribution of particles over the posterior distribution.

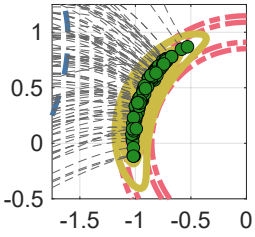


Fig. 3: Particle flow induced by ODE solution without measurement perturbation

B. Stochastic particle flow

As an alternative approach, we can write down the following Itô stochastic differential equation (SDE):

$$dx = f(x,\tau)\ d\tau + B(x,\tau)\ dw \tag{41}$$

where

$$f(x,\tau) = P\tilde{H}^T R^{-1}(y - h(x)) \tag{42}$$

is the *drift term* and

$$B(x,\tau) = P\tilde{H}^T R^{-1/2} \tag{43}$$

is the *diffusion term*. B is defined such that $Q = BB^T$, where

$$Q = P\tilde{H}^T R^{-1}\tilde{H}P \tag{44}$$

Using the Euler-Maruyama method, we define the following particle flow:

1) Compute the prior mean, m, and the sample covariance, P, of the particles using (34).
2) Solve Eqs. (25)-(26) with initial conditions m and P. Obtain time series $\left\{\Delta\tau^{(1)} \ \ldots \ \Delta\tau^{(n_t)}\right\}$.
3) For each time step, perform the following update for each particle:

$$x_i^{(k+1)} = x_i^{(k)} + P^{(k)}\tilde{H}_i^{(k)} R^{-1}\left(y - h\left(x_i^{(k)}\right)\right) d\tau + P^{(k)}\tilde{H}_i^{(k)} R^{-1/2} dw \tag{45}$$

where

$$d\tau = \Delta\tau^{(k)} \tag{46}$$

$$dw \sim \mathcal{N}(0, \Delta\tau^{(k)} I) \tag{47}$$

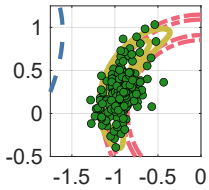
4) After all particles have undergone one update step, recompute the sample covariance.

$$P^{(k+1)} = \frac{1}{n_p - 1}\sum_{i=1}^{n_p}\left(x_i^{(k+1)} - m\right)\left(x_i^{(k+1)} - m\right)^T \tag{48}$$

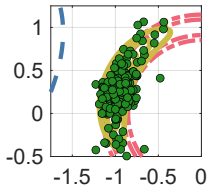
5) Repeat steps 3-4 until all time steps are exhausted.

Euler-Maruyama is a popular stochastic integration method because it is easy to implement. For insight into other practical stochastic integration techniques for particle flow, see Ref. [22]. Ref. [23] is a wonderful introduction to SDEs for readers familiar with state estimation.

Fig. 4a shows the result of this method for the range measurement example, this time for 200 particles. Lines showing the particles' individual trajectories are not included, since stochastic flows are not smooth. While the Euler-Maruyama results take advantage of the adaptive time steps for the ODE solution, this is not strictly correct, as time-steps have to be adaptively choosen for the SDE. Fig. 4b shows the particle flow results using a strong order 1.5 adaptive SDE solver, Random Sampling with Memory (RSwM1) [18]. Euler-Maruyama has strong order 0.5 [23].



(a) Euler-Maruyama



(b) RSwM1

Fig. 4: Particle flow induced by SDE solution

All the particles move through the stochastic flow together, as the sample covariance is used in (42)-(43). Alternatively, the theoretical covariance (38) may be used in (45). This has the advantage of making the flow parallelizable, like the ODE flow. However, the sample covariance—likely the best available representation of the uncertainty after each update step—goes unused after the first step.

C. Note on multimodal posterior distributions

Consider again the range measurement $y = 1$. This time, the prior state estimate is $x = \begin{bmatrix}0 & 0\end{bmatrix}^T$, and the prior covariance matrix is $P = \left[\begin{smallmatrix}1 & 0\\ 0 & 0.05\end{smallmatrix}\right]$. Placing the prior mean at the origin yields a bimodal posterior distribution. Fig. 5 shows the result of the ODE flow for a set of 500 particles (see Sec. III-A).

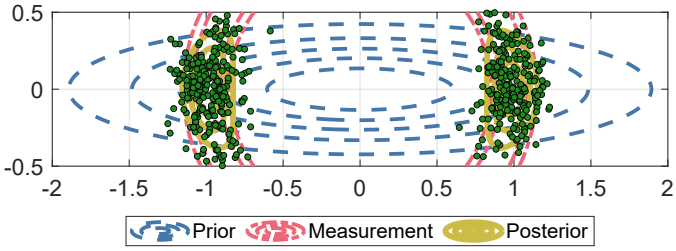


Fig. 5: Result of ODE flow for bimodal posterior example

Since the measurement Jacobian (29) is undefined for $x = \begin{bmatrix}0 & 0\end{bmatrix}^T$, a small perturbation was added to the prior state estimate in order to generate the time steps $\Delta\tau^{(k)}$. Clearly, the ODE flow is able to move particles to both areas of high probability in the posterior distribution.

The flow is computed by applying a BRUF update to each particle. For each particle, the initial BRUF state estimate is the *location of the particle itself*; not the sample mean. By contrast, every BRUF is initialized with the sample covariance. This means that the posterior distribution looks slightly different for each particle. In other words, while Fig. 5 shows a posterior distribution with two identical "lobes" for prior

state estimate $x = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$, these lobes will be lopsided for particles initially located away from the origin. The particles travel to the lobe closest to where they started.

IV. Lorenz '63 with Range and Angles

We demonstrate the performance of the proposed particle flow filters on a well-known chaotic dynamical system with high-precision nonlinear position measurements. The Lorenz '63 dynamics are [24]:

$$\begin{aligned} \dot{x}_1 &= \sigma(x_2 - x_1) \\ \dot{x}_2 &= x_1(\rho - x_3) - x_2 \\ \dot{x}_3 &= x_1 x_2 - \beta x_3 \end{aligned} \tag{49}$$

where $\sigma = 10$, $\rho = 28$, and $\beta = \frac{8}{3}$ yield the familiar butterfly trajectory. Fig. 6 shows the evolution of the trajectory for the first 40 time units.

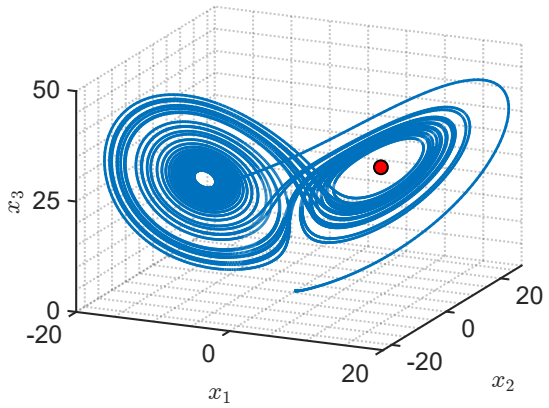


Fig. 6: Lorenz '63 dynamics

An instrument placed at the equilibrium point $\begin{bmatrix} 6\sqrt{2} & 6\sqrt{2} & 27 \end{bmatrix}^T$ measures range, azimuth, and elevation to the target. These quantities can be defined in terms of the line-of-sight vector

$$\vec{r} = x - \begin{bmatrix} 6\sqrt{2} & 6\sqrt{2} & 27 \end{bmatrix}^T \tag{50}$$

where the notation $\vec{(\cdot)}$ is used to differentiate the vector-valued line-of-sight from the range value

$$r = \|\vec{r}\|. \tag{51}$$

The angles are

$$\begin{aligned} \alpha &= \tan^{-1}(\vec{r}_2/\vec{r}_1) \\ \epsilon &= \sin^{-1}(\vec{r}_3/r) \end{aligned} \tag{52}$$

The measurement model is

$$\begin{aligned} y &= h(x) + \nu \\ \nu &\sim \mathcal{N}(0, R) \end{aligned} \tag{53}$$

where

$$h(x) = \begin{bmatrix} r & \alpha & \epsilon \end{bmatrix}^T \tag{54}$$

and

$$R = \begin{bmatrix} 0.1^2 & 0 & 0 \\ 0 & 0.01^2 & 0 \\ 0 & 0 & 0.01^2 \end{bmatrix}. \tag{55}$$

All angular values are given in radians.

The measurement Jacobian is

$$\tilde{H} = \begin{bmatrix} \vec{r}_1/r & \frac{-\vec{r}_2}{\vec{r}_1^2[1+(\vec{r}_2/\vec{r}_1)^2]} & \frac{-\vec{r}_1\vec{r}_3}{r^3\sqrt{1-(\vec{r}_3/r)^2}} \\ \vec{r}_2/r & \frac{1}{\vec{r}_1[1+(\vec{r}_2/\vec{r}_1)^2]} & \frac{-\vec{r}_2\vec{r}_3}{r^3\sqrt{1-(\vec{r}_3/r)^2}} \\ \vec{r}_3/r & 0 & \frac{1}{\sqrt{1-(\vec{r}_3/r)^2}}\left[\frac{1}{r} - \frac{\vec{r}_3^2}{r^3}\right] \end{bmatrix}^T \tag{56}$$

Each filter in this study was initialized with a set of particles $X(0) = \{x_1(0) \dots x_{n_p}(0)\}$ drawn from

$$X(0) \sim \mathcal{N}(x(0), P(0)) \tag{57}$$

where $x(0)$ is the true initial state

$$x(0) = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T \tag{58}$$

and

$$P(0) = I_{3\times 3}. \tag{59}$$

The filter dynamics (49) were integrated using a fourth-order Runge Kutta scheme. A measurement was received every 0.12 time units. The state estimate at time t is simply the mean of the particles:

$$\hat{x}(t) = \frac{1}{n_p}\sum_{i=1}^{n_p} x_i(t). \tag{60}$$

Five filters were tested on this system: the ODE flow (Section III-A), the stochastic flow induced by Euler-Maruyama (Section III-B), the stochastic flow induced by RSwM1 (Section III-B), Gromov flow, and Daum-Huang exact flow. The Gromov flow was induced by Euler-Maruyama, like the SDE flow [22]. Daum-Huang exact flow is a diffusion-free flow (i.e., $B(x,t) = 0$ in (41)). A skeleton of the algorithm used to implement the exact flow is given in Ref. [25], Alg. 2. In place of the usual exact flow equations, we used the version in [26] (see Supplementary Material), which includes a correction for the linearization error. In the SDE flow and the RSwM1 solver, the theoretical covariance (38) was used in place of the sample covariance after the first update step.

Gromov flow and Daum-Huang exact flow are sometimes run with a companion EKF or UKF, which provide the propagated covariance matrix required by the flow equations [3]. We did not include any companion filters in this work. Instead, each measurement update begins with the sample covariance (34) of the propagated particles. This can be problematic given the deterministic dynamics (49); the particles tend to spread out in a straight line, resulting in near-zero eigenvalues in the sample covariance matrices. As a regularization, the matrix $0.01 \cdot I_{3\times 3}$ is added to the sample covariance in all the filters at the beginning of each measurement update. We do not resample.

Gromov flow and Daum-Huang flow were executed with 50 update steps of uniform size $\Delta\tau^{(k)} \triangleq 1/50$. For the proposed ODE particle flow and the Euler-Mauryama SDE solution, the ODE solver averaged about 45 steps per update. The adaptive SDE solver, RSwM1, averaged about 15 steps per measurement update, though a large number of random samples may be generated per step.

Fig. 7 shows the root-sum-of-squares (RSS) positioning error for 50 Monte Carlo runs with 1000 measurement updates each. All filters were executed with 25 particles. The RSS error is defined as

$$\mathrm{RSS}(t) = \sqrt{\sum_{n=1}^{n_m} \|\varepsilon_n(t)\|^2} \tag{61}$$

where n_m is the number of Monte Carlo runs, and $\varepsilon(t)$ is the total position error at time t:

$$\varepsilon(t) = \hat{x}(t) - x(t). \tag{62}$$

Clearly, certain time steps are prone to large positioning error. The proposed filters are better able to handle these difficult measurements than Gromov flow and Daum-Huang exact flow.

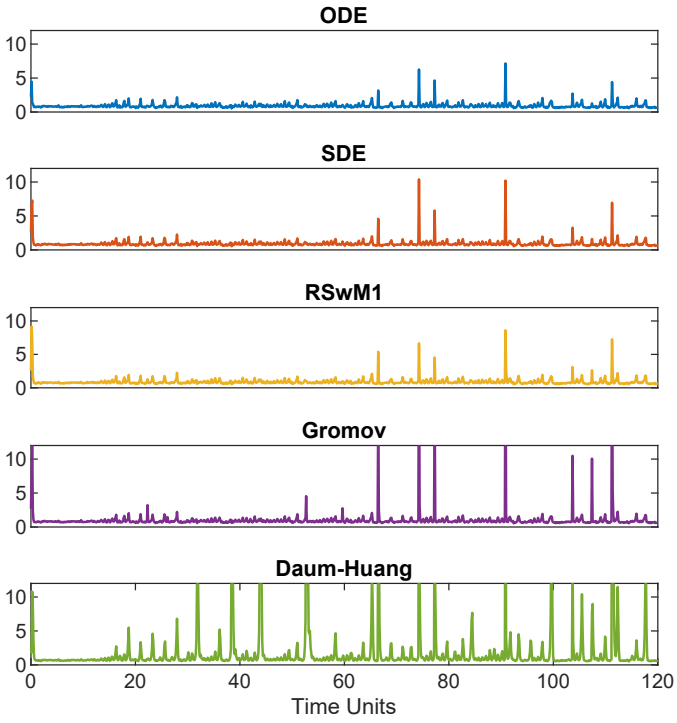


Fig. 7: RSS error for filters with 25 particles

Table I shows the mean spatio-temporal RMSE results for filters with 10, 15, and 25 particles. The mean spatio-temporal RMSE is:

$$\mathrm{RMSE} = \frac{1}{n_m} \sum_{n=1}^{n_m} \sqrt{\frac{1}{3}\frac{1}{n_t} \sum_{j=1}^{n_t} \|\varepsilon_n(t)\|^2}. \tag{63}$$

The ODE flow has the best performance in all three cases. The RSwM1 solution and the Euler-Maruyama SDE solution are close behind, followed by Gromov flow and finally Daum-Huang. The outstanding performance of the ODE flow likely reflects the stability of the ODE solution; a deterministic solution with perturbed measurements is able to match the posterior distribution without relying on randomly-generated diffusion terms in each update step.

TABLE I: RMSE Results for Lorenz '63

	ODE	**SDE**	**RSwM1**	**Gromov**	**DH**
$n_p = 10$	**0.085**	0.097	0.092	0.184	0.420
$n_p = 15$	**0.083**	0.092	0.090	0.178	0.418
$n_p = 25$	**0.082**	0.091	0.088	0.179	0.418
$n_p = 100$	**0.080**	0.090	0.086	0.174	0.418

V. Conclusion

This work presents an ODE formulation of the nonlinear measurement update. We show that the discrete-time measurement can be expressed in differential form by choosing an inflated measurement noise covariance as the power spectral density of a continuous-time measurement. We recover the measurement update terms in the Kalman-Bucy filter. We then use this ODE formulation to induce a particle flow. We show the results of this particle flow for a single range measurement with a Gaussian prior distribution.

We then present two filtering algorithms; a deterministic flow with perturbed measurements and a stochastic flow. Each filter only requires one ODE solve per measurement update. After the ODE is solved at the particle mean, the pseudotime time series is used to flow the rest of the particles. In the ODE formulation, each particle undergoes a BRUF update in parallel. The computational complexity of the update (post-ODE solve) essentially matches that of comparable particle flow filters; an EKF update is performed for each particle at each update step.

The SDE formulation is more open-ended; like ODEs, many solvers exist. We present an algorithm that uses Euler-Maruyama with the pseudotime series from an ODE solver. We also test the approach with a recently-introduced SDE solver, Rejection Sampling with Memory (RSwM1). One potential avenue for future work is the use of statistical linearization instead of local linearization about each particle. If all the particles move through the flow together, then statistical techniques may be employed to compute the Kalman gain at each step.

Our filtering results show that the ODE flow filter produces the best estimation results for a difficult problem with chaotic dynamics and a highly accurate nonlinear measurement. The three filters introduced in this work outperform two existing particle flow filters given the same average number of pseudotime steps per update.

Section III shows that diffusion is not required to move particles onto the the Bayesian posterior. Instead, by generating a set of perturbed measurements, the diffusion-free ODE flow can produce a healthy distribution of particles over the posterior distribution.

While efforts were made reduce the computational complexity of the particle flow filters in this work (i.e., only performing a single ODE solve at measurement time), in the future, we would like to improve the efficiency of the ODE solution itself. The covariance matrix must be integrated in pseudotime with the state vector. This means that an entire covariance matrix

enters the ODE solver with the state vector, for a total of $n+n^2$ terms.

A first obvious solution is only including the terms in the lower or upper triangle of the covariance matrix, though rebuilding the covariance matrix from these terms for each solution step proved to be computationally burdensome. Surely much more efficient solution methods exist. By reducing the computational complexity of the ODE solution, we could extend the particle flow filters proposed in this work to systems with higher-dimensional state spaces.

REFERENCES

[1] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter: Particle filters for tracking applications*. Artech house, 2003.

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

[3] S. Choi, P. Willett, F. Daum, and J. Huang, "Discussion and application of the homotopy filter," in *Signal Processing, Sensor Fusion, and Target Recognition XX*, vol. 8050, pp. 734–745, SPIE, 2011.

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

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

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

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

[8] K. Michaelson, A. A. Popov, and R. Zanetti, "Recursive update filtering: A new approach," in *Proceedings of the 33rd AAS/AIAA Space Flight Mechanics Meeting*, 2023.

[9] N. Chopin, "A sequential particle filter method for static models," *Biometrika*, vol. 89, no. 3, pp. 539–552, 2002.

[10] M. A. Skoglund, G. Hendeby, and D. Axehill, "Extended Kalman filter modifications based on an optimization view point," in *2015 18th International Conference on Information Fusion (Fusion)*, pp. 1856–1861, IEEE, 2015.

[11] U. D. Hanebeck and J. Steinbring, "Progressive Gaussian filtering based on dirac mixture approximations," in *2012 15th International Conference on Information Fusion*, pp. 1697–1704, IEEE, 2012.

[12] A. A. Emerick and A. C. Reynolds, "Ensemble smoother with multiple data assimilation," *Computers & Geosciences*, vol. 55, pp. 3–15, 2013.

[13] K. Michaelson, A. A. Popov, and R. Zanetti, "Ensemble Kalman filter with Bayesian recursive update," in *2023 26th International Conference on Information Fusion (FUSION)*, pp. 1–6, IEEE, 2023.

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

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

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

[17] L. F. Shampine and M. W. Reichelt, "The MATLAB ODE suite," *SIAM journal on scientific computing*, vol. 18, no. 1, pp. 1–22, 1997.

[18] C. Rackauckas and Q. Nie, "Adaptive methods for stochastic differential equations via natural embeddings and rejection sampling with memory," *Discrete and continuous dynamical systems. Series B*, vol. 22, no. 7, p. 2731, 2017.

[19] J. Havlík and O. Straka, "Performance evaluation of iterated extended Kalman filter with variable step-length," in *Journal of Physics: Conference Series*, vol. 659, p. 012022, IOP Publishing, 2015.

[20] 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, pp. 10143–10162, 1994.

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

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

[23] S. Särkkä and A. Solin, *Applied stochastic differential equations*, vol. 10. Cambridge University Press, 2019.

[24] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of atmospheric sciences*, vol. 20, no. 2, pp. 130–141, 1963.

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

[26] S. Pal, L. Ma, Y. Zhang, and M. Coates, "RNN with particle flow for probabilistic spatio-temporal forecasting," in *International Conference on Machine Learning*, pp. 8336–8348, PMLR, 2021.

APPENDIX

In previous work, we used the information-form measurement update to prove that the BRUF update matches the Kalman update in the linear case [13], [14]. The *information state* is

$$z = P^{-1}x \tag{64}$$

where P^{-1} is the *information matrix*; the inverse of the covariance matrix. For linear systems, the information update is

$$\begin{aligned} \hat{z} &= z + H^T R^{-1} y \\ \hat{P}^{-1} &= P^{-1} + H^T R^{-1} H. \end{aligned} \tag{65}$$

Applying the BRUF step α_i to the *nonlinear* information update, we can define the recursion

$$\begin{aligned} \hat{z}^{(i+1)} &= \hat{z}^{(i)} + \alpha_i \tilde{H}^T R^{-1} y \\ P_{i+1}^{-1} &= P_i^{-1} + \alpha_i \tilde{H}^T R^{-1} \tilde{H}. \end{aligned} \tag{66}$$

Using the same logic as (23)-(24), we can now write down the differential equations:

$$\begin{aligned} \dot{z} &= \tilde{H}^T R^{-1} y \\ \dot{P}^{-1} &= \tilde{H}^T R^{-1} \tilde{H}. \end{aligned} \tag{67}$$

We return to the state space beginning with the information matrix. If $PP^{-1} = I$, then

$$(P\dot{P}^{-1}) = \dot{P}P^{-1} + P(\dot{P^{-1}}) = \dot{I} = 0. \tag{68}$$

Then,

$$\begin{aligned} \dot{P} &= -P(\dot{P^{-1}})P \\ &= -P\tilde{H}^T R^{-1} \tilde{H} P. \end{aligned} \tag{69}$$

For the state vector, $x = Pz$. Therefore,

$$\begin{aligned} \dot{x} &= \dot{P}z + P\dot{z} \\ &= -P\tilde{H}^T R^{-1} \tilde{H} P P^{-1} x + P\tilde{H}^T R^{-1} y \\ &= P\tilde{H}^T R^{-1}(y - \tilde{H}x) \end{aligned} \tag{70}$$

Taking $\tilde{H}x \approx h(x)$, we recover (25)-(26).