Projected Feedback Particle Filtering for Chaotic Dynamical Systems Using Lyapunov Vectors

Yujing Zhou
Mechanical and Aerospace Engineering
Princeton University
Princeton, New Jersey, US
yz1324@princeton.edu

Ryne Beeson
Mechanical and Aerospace Engineering
Princeton University
Princeton, New Jersey, US
ryne@princeton.edu

***Abstract*—Particle flow methods are effective in resolving the particle degeneracy issue in the standard particle filtering algorithm. However, flow methods have their own difficulties, such as the necessity to solve a Poisson equation in the feedback particle filtering (FPF) method. This is computationally heavy, and we observe a numerical sensitivity and singularity issue dependent on parameter selection when applying to chaotic dynamical systems with limited particle size and coarse integration step size. In this paper, we address the numerical singularity issue by flowing particles in the unstable subspace (UAS), and we name the novel method the projected FPF. It brings the local dynamical information into the assimilation step by using the finite-time Lyapunov exponents and vectors to project observations and particle states to the UAS, where the error diverges. The projected FPF is tested against the Lorenz 1963 model – a nonlinear, low-dimensional, chaotic dynamical system.**

***Index Terms*—data assimilation, particle filters, particle flow, Lyapunov vector, chaotic dynamics, Lorenz '63 model**

I. Introduction

Data assimilation (DA) or filtering is concerned with finding the posterior distribution, a conditional probability distribution for an unobserved signal process given indirect observations. We are interested in such problems when applying to the geoscience regime. Current DA methods for weather prediction include the ensemble Kalman filter and 4D-var, but they both assume linear Gaussian updates when assimilating observations, which is losing its soundness as the physical model becomes inherently non-linear and non-Gaussian when modeling the convective process [1]. Additionally, the physical model is usually chaotic and high-dimensional; signal dimension can be of order $\mathcal{O}(10^9)$ and the observation of order $\mathcal{O}(10^7)$ in a single day. Thus, more general DA techniques are required for higher assimilation accuracy.

Particle filtering is a sequential Monte Carlo method that uses weighted particles to approximate the posterior distribution, which makes it absolutely general. However, the particle degeneracy issue prevents its application to high-dimensional problems, meaning that after several iterations into the simulation, all but one particles have weights close to zero. Thus, useful statistical information is lost. It is caused by the pointwise multiplication between a proposal distribution and the observation likelihood. A sub-optimal but commonly used proposal distribution is the prior distribution which does not well represent the observation. It has been proved that the particle size scales exponentially with the dimension of observations to maintain the effective sampling size [2], [3]. The nudged particle filter (nPF) method [4], [5] aims to find better proposal distributions by bringing particles closer to the observation through a control term, but it does not inherently solve the particle degeneracy issue.

In this paper, we consider a flow-based approach that was first proposed by Daum and Huang in [6]–[9]. Instead of updating particles' weights according to Bayes' rule, the particle flow method moves the particle ensemble from the prior to the posterior states while keeping the weights static. It is achieved by solving an ordinary differential equation (ODE) within a pseudo-time parameterized by λ at the assimilation step. Thus, the particle degeneracy issue is circumvented. To the best of our knowledge, the particle-flow based approaches have yet to be applied to high dimensional systems such as those that arise in geoscience or chaotic systems, where particle degeneracy is made worse. We are motivated to apply such a method in these situations.

Over the past decade, a series of control-based particle filtering methods, called the feedback particle filter (FPF), were introduced to solve the particle degeneracy issue [10]–[12]. The continuous-discrete time setup [11] fits the most to geoscience problems because of the discrete sparse observations, and it leverages the particle-flow idea. Under the FPF framework, each particle is governed by the original signal dynamics with an additional control term. The control is said to be optimal if in the mean-field limit the conditional density of the particle ensemble given observations is the same as the posterior density. Instead of using Bayes' rule, observations are assimilated by computing a control term in FPF, leaving the particle weights unchanged. However, the drawback of FPF is twofold. First, the optimal control law requires solving a Poisson equation [13], which is computationally heavy. We refer interested readers to [12] and [14] for detailed discussions on numerical approximations for the Poisson equation. Second, when applying to chaotic systems, the posterior distribution does not have to be supported on the chaotic attractor, which did not show up when testing on the linear system in [11]. We noticed that with aggressive parameter selections (i.e., limited number of particles, coarse basis approximation for

the Poisson equation, and larger pseudo-time step size), the FPF suffered numerical singularity issues. In particular, the posterior particle states can blowup to infinity.

Therefore in this paper, we suggest the use of particle flow in the unstable subspace (UAS) to mitigate two issues. First, reducing the dimension of the Poisson problem that needs to be solved for the control gain function. Second, alleviating the physical non-realizability issue (i.e., the posterior is not supported on the chaotic attractor). The approach taken in this paper is driven by the work from [15], which uses finite-time Lyapunov vectors, left-singular vectors, and future right-singular vectors to perform assimilation in the UAS, yielding higher effective sampling size. This idea is originated from [16] and [17] that dealt with deterministic signal processes with low observation noises in the linear Gaussian case.

The proposed algorithm is tested against the Lorenz 1963 model [18], which is a low-dimensional, nonlinear, and chaotic system, meaning that a small initial perturbation grows exponentially in time. Despite having deterministic dynamics, its behavior is non-periodic and appears random, making it a common testbed for geoscience filtering algorithms. The Lorenz '63 model will be formally introduced in Section IV.

This paper is organized as follows: Section II discusses the particle degeneracy issue and the particle flow method. Then in Section III we review the background on Lyapunov exponents and vectors. We introduce the idea of flow in the UAS in Section IV and demonstrate the filtering results in Section V. We finish with a conclusion in Section VI.

II. Particle Flow

In this section, we first state the particle degeneracy issue for particle filtering when applied to high-dimensional problems. We then recount how the FPF algorithm from [11] inherently solves this issue.

A. Particle Degeneracy

Consider the following continuous-time signal process with discrete-time observation process:

$$\begin{aligned} \mathrm{d}\mathbf{X}_t &= \mathbf{b}\left(\mathbf{X}_t\right)\mathrm{d}t + \boldsymbol{\sigma}\left(\mathbf{X}_t\right)\mathrm{d}\mathbf{W}_t, \qquad & \mathbf{X}_0 &= \mathbf{x} \in \mathbb{R}^m, \\ \mathbf{Y}_{t_k} &= \mathbf{h}\left(\mathbf{X}_{t_k}\right) + \boldsymbol{\xi}_{t_k}, & \mathbf{Y}_0 &= \mathbf{0} \in \mathbb{R}^d, \end{aligned} \tag{1}$$

where $\mathbf{X}$ is the signal process, $\mathbf{W}$ is a standard Brownian motion taking values in $\mathbb{R}^m$, $\mathbf{b}$ is the drift term, σ is the dispersion coefficient, $(\mathbf{Y}_{t_k})_{k\in\mathbb{N}}$ is a sequence of discrete observations, $\mathbf{h}$ the sensor function, and $(\boldsymbol{\xi}_{t_k})_{k\in\mathbb{N}}$ is a sequence of independent Gaussian random variables with zero mean and covariance $\mathbf{R}_y \in \mathbb{R}^d$. We assume that $\mathbf{W}$ and $\boldsymbol{\xi}$ are independent.

The goal in filtering is to find the posterior distribution, which we denote as $\pi_{k|k}$, a conditional distribution of $\mathbf{X}_k$ given all observations up to and including the time t_k. Since $(\mathbf{Y}_{t_k})$ is a Markov process in our setup, only the last observation is required in the posterior solution. Particle filtering in this setting is a sequential method and the prior distribution can be found by the Chapman-Kolmogorov equation:

$$\pi_{k+1|k}(\mathbf{x} \mid \mathbf{Y}) \equiv \int_{\mathbb{R}^m} \rho_{k+1|k}\left(\mathbf{x} \mid \mathbf{x}'\right)\pi_{k|k}\left(\ \mathrm{d}\mathbf{x}' \mid \mathbf{Y}\right), \tag{2}$$

In (1), $\rho_{k+1|k}$ is the probability transition function from t_k to t_{k+1}. The posterior distribution at t_{k+1} can then be calculated from Bayes' formula:

$$\pi_{k+1|k+1}\left(\mathbf{x} \mid \mathbf{Y}\right) = \frac{f_{k+1|k+1}\left(\mathbf{Y} \mid \mathbf{x}\right)\pi_{k+1|k}\left(\mathbf{x} \mid \mathbf{Y}\right)}{\int_{\mathbb{R}^m} f_{k+1|k+1}\left(\mathbf{Y} \mid \mathbf{x}\right)\pi_{k+1|k}\left(\ \mathrm{d}\mathbf{x} \mid \mathbf{Y}\right)},$$

where $f_{k|k}(\mathbf{Y}|\mathbf{x})$ is the likelihood of $\mathbf{Y}_k$ given $\mathbf{x}_k$.

Particle filtering represents the posterior and prior distributions with a finite convex combination of Dirac distributions. For example, let $\pi_{0|0}$ represents the initial posterior distribution. Approximating this distribution with $N \in \mathbb{N}$ particles, is equivalent to,

$$\pi_{0|0}(x) \approx \sum_{i=1}^{N} w_0^i \delta(x - x_0^i), \tag{3}$$

where $(w_0^i)_{i=1}^N$ are initial particle weights, each of which satisfies $w_0^i \in [0,1]$, $(x_0^i)_{i=1}^N$ are the initial location of each particle in $\mathbb{R}^m$, and δ is a Dirac distribution[1]. We would like to sample from the posterior distribution $\pi_{k|k}$ to calculate such statistics as the mean and covariance, but this often cannot be done directly. In the sequential importance sampling (SIR) framework, one assumes that they can sample from a proposal distribution q, which is absolutely continuous with respect to π, with density function (also called the importance weight function) α. Writing this out for the posterior distribution at time t_{k+1}, we have,

$$\pi_{k+1|k+1}\left(\mathbf{x} \mid \mathbf{Y}\right) = \alpha_{k+1}(x) q_{k+1}(\mathbf{x}). \tag{4}$$

Although not optimal, the most common choice of the proposal distribution q is the prior distribution in (1). Then using a Monte Carlo (MC) approximation for the proposal (i.e., following the form of (3)), we get that

$$\alpha_k(x) = f_{k+1|k+1}(\mathbf{Y}|\mathbf{x}),$$

and (4) becomes:

$$\pi_{k+1|k+1}(\mathbf{x}|\mathbf{Y}) \propto \sum_{i=1}^{N} w_k^i f_{k+1|k+1}(\mathbf{Y}|\mathbf{x})\delta(\mathbf{x} - \mathbf{x}_{k+1}^i). \tag{5}$$

Due to the randomness in (1) from the Brownian motion and Gaussian observation noise, the finite collection of prior particles may not represent the observation well. Hence, the updated weight for most particles, $w_{k+1}^i = w_k^i f_{k+1|k+1}(\mathbf{Y}|\mathbf{x})$, will eventually be close to zero. This is the issue of particle degeneracy (collapse, impoverishment etc.).

B. Particle Flow in the FPF algorithm

In the SIR algorithm, particles are transformed from the prior to the posterior in one step by updating their weights with (5). In contrast, the particle flow method achieves this goal by introducing a pseudo-time parameterized by $\lambda \in [0,1]$, in which particles are moved "smoothly" from the prior to the posterior by solving an ODE. In particular, consider the posterior distribution from Bayes' rule: $f(x) = g(x)h(x)$, where

[1] note that we use superscripts for indexing the particles and subscripts for indexing time

we ignore the normalization term, and $g(x)$ and $h(x)$ are respectively the prior and likelihood distributions. After a log-transformation, we have: $\log(f(x) = \log(g(x)) + \log(h(x))$. Then, we may define a homotopy function $\hat{f}(\lambda, x)$ such that:

$$\log(\hat{f}(\lambda, x)) = \log(g(x)) + \lambda \log(h(x)), \tag{6}$$

where $\lambda \in [0, 1]$, and $\hat{f}(0, x) = g(x)$ is the prior while $\hat{f}(1, x) = g(x)h(x)$ is the posterior. The homotopy function is then normalized and differentiated with respect to λ, yielding the evolution of the distribution of $\mathbf{X}_k$ within the pseudo-time, denoted as $\frac{\partial \rho_k^*}{\partial \lambda}(x, \lambda)$.

Yang combines the particle flow idea with the FPF regime in [11] by designing the particle state $S_k^i(\lambda)$ within the pseudo-time to follow the ODE:

$$\frac{\mathrm{d}S_k^i}{\mathrm{d}\lambda}(\lambda) = \mathrm{K}\left(S_k^i(\lambda), \lambda\right) Y_k + u\left(S_k^i(\lambda), \lambda\right), \tag{7}$$

where $\mathrm{K}(x, \lambda) \in \mathbb{R}^{m \times d}$ and $u(x, \lambda) \in \mathbb{R}^m$ are control terms to be chosen. With $S_k^i(0)$ being the prior particle states, $\{\mathrm{K}(x, \lambda), u(x, \lambda)\}$ is said to be optimal if it moves particles to $S_k^i(\lambda)$, whose distribution coincides with $\rho_k^*(x, \lambda)$.

The optimal pair $\{\mathrm{K}(x, \lambda), u(x, \lambda)\}$ has been proved to obtain the following form. For each $\lambda \in [0, 1]$, the gain function $\mathrm{K} = [\nabla\phi_1^T, \nabla\phi_2^T, ..., \nabla\phi_d^T]$ is obtained as a solution to the following Poisson equation:

$$\begin{aligned} \nabla \cdot (\rho \nabla \phi_j) &= -\left(h_j - \hat{h}_j\right) \rho, \\ \int \phi_j(x, \lambda) \rho(x, \lambda) \mathrm{d}x &= 0, \end{aligned} \tag{8}$$

for $j = 1, ..., d$. h_j is the j-th component from the sensor function, $\hat{h}_j = \sum_{i=1}^N h_j(S^i(\lambda))$, and ρ is the distribution of $S^i(\lambda)$. The function u has the form of: $u(x, \lambda) = -\frac{1}{2}\mathrm{K}(x, \lambda)(h(x) + \hat{h}) + \frac{1}{2}\Omega(x, \lambda)$, where $\Omega = \nabla\psi$ is a solution to the following Poisson equation:

$$\begin{aligned} \nabla \cdot (\rho \nabla \psi) &= -(g - \hat{g})\, \rho, \\ \int \psi(x, \lambda) \rho(x, \lambda) \mathrm{d}x &= 0, \end{aligned} \tag{9}$$

where $g := \sum_{j=1}^d \nabla\phi_j \cdot \nabla h_j^T$ and $\hat{g} := |\hat{h}|^2 - \widehat{|h|^2}$. Eq. (7) can be written as

$$\frac{\mathrm{d}S_k^i}{\mathrm{d}\lambda}(\lambda) = \mathrm{K} \underbrace{\left(Y_k - \frac{1}{2}[h(S_k^i(\lambda)) + \hat{h}(\lambda)]\right)}_{I_k^i(\lambda)} + \frac{1}{2}\Omega, \tag{10}$$

In this paper, constant approximations to (8) and (9), meaning that $\tilde{\mathrm{K}} = \mathbb{E}[\mathrm{K}]$ and $\tilde{\Omega} = \mathbb{E}[\Omega]$, are used:

$$\begin{aligned} \tilde{\mathrm{K}} &\approx \frac{1}{N} \sum_{i=1}^N S_n^i(\lambda) \left(h\left(S_n^i(\lambda)\right) - \hat{h}\right)^T \\ \tilde{\Omega} &\approx \frac{1}{N} \sum_{i=1}^N S_n^i(\lambda) \left(g\left(S_n^i(\lambda)\right) - \hat{g}\right). \end{aligned} \tag{11}$$

Interested readers are referred to [12] for the Galerkin method and [14] for the diffusion-map method to numerically solve (8) and (9).

In summary, posterior particles at t_k are propagated to the prior states at t_{k+1} by following the signal process defined in (1). Then, with observations available at t_{k+1}, prior particles are updated by (7) within the pseudo-time to arrive at the posterior states.

Through this process, particle weights remain unchanged, which overcomes the particle degeneracy issue. However, it involves numerical challenges. First of all, solving two Poisson equations in (8) and (9) is computationally demanding in high-dimensional problems. Additionally, in nonlinear chaotic systems, state variables reside on a lower dimensional attractor, but the particle state updates within the pseudo-time, given by (7), (8), and (9), does not depend on the signal process, in particular the drift term $\mathbf{b}$. Therefore, particles can end up being not physically realizable. Depending on the choice of the step size $\Delta\lambda$, the pseudo-update can also become unstable, leading particle states to blowup. We would like to approach these issues by bringing local dynamical information into the pseudo-time update through the finite-time Lyapunov exponents and vectors.

III. Lyapunov Exponents and Vectors

Lyapunov exponents and their associated vectors provide key information of a dynamical system's stability and sensitivity to perturbations. Their finite-time variants split the tangent space at final time to stable, unstable, or neutral subspaces.

Consider a dynamical system defined by the flow map $\boldsymbol{\varphi} : M \times \mathbb{R} \to M$ on a smooth Riemannian manifold M. The tangent bundle $TM \equiv \coprod_{\mathbf{p} \in M} T_{\mathbf{p}}M$ is the disjoint union of the tangent spaces $T_{\mathbf{p}}M$ at each point $\mathbf{p} \in M$. $\mathbf{v} \in T_{\mathbf{p}}M$ is the tangent vector. Denote the differential of flow map from zero to t as $\boldsymbol{D\varphi}_t$. Then, the Lyapunov exponent at $(\boldsymbol{p}, \mathbf{v})$ is defined as follows:

$$\begin{aligned} &\lambda : TM \longrightarrow \mathbb{R} \\ &(\mathbf{p}, \mathbf{v}) \longmapsto \lambda(\mathbf{p}, \mathbf{v}) \equiv \lim_{t \to \infty} \frac{1}{|t|} \ln \|\boldsymbol{D\varphi}_t(\mathbf{p})\mathbf{v}\|, \end{aligned} \tag{12}$$

where $\boldsymbol{D\varphi}_t(\mathbf{p})\mathbf{v}$ moves tangent vectors from $T_{\mathbf{p}}M$ to $T_{\boldsymbol{\varphi}_t(\mathbf{p})}M$.

If $(\boldsymbol{\varphi}, M, \mu)$ is an ergodic dynamical system with μ an ergodic measure on M for which φ, a diffeomorphism, is a measure preserving map, then Oseledet's multiplicative ergodic theorem [19] states that on a finite-dimensional space, we can characterize the behavior of infinitesimal perturbations for almost every point using finite number of constant values and tangent spaces.

In applications, the finite-time Lyapunov exponents and vectors are often more relevant. The finite-time variants of (12) is given by:

$$\lambda_t(\mathbf{p}, \mathbf{v}) \equiv \frac{1}{|t|} \ln \|\boldsymbol{D\varphi}_t(\mathbf{p})\mathbf{v}\|. \tag{13}$$

However, we no longer have Oseledet's theorem, and the finite-time Lyapunov exponent depends on the base point. In practice, the finite-time Lyapunov exponents and vectors can be calculated using a modified Gram-Schmidt procedure based on the QR decomposition [20]. To start off, an orthogonal set of basis vectors Q_0 is integrated a small step from $t = 0$ to $t = \delta_t$ under the tangent linear dynamics, yielding $\boldsymbol{D}\boldsymbol{\varphi}_{\delta_t}$. Then, a QR decomposition of $\boldsymbol{D}\boldsymbol{\varphi}_{\delta_t}$ is performed to obtain an orthogonal matrix Q_1 and an upper triangular matrix R_1. Next, Q_1 is set as the new basis matrix, and the same procedure is repeated until the final time. Now, the finite-time Lyapunov vectors can be calculated based on the history of R_i and the orthogonal matrix Q^* gives the associated Lyapunov vectors. Based on the sign of each finite-time Lyapunov exponent, the Lyapunov vectors split the tangent space at final time to stable, neutral, and unstable subspaces.

IV. Projected Particle Flow on the Lorenz 1963 Model

In this section, we introduce the novel projected FPF method, which enables particle flow in the UAS, aiming to improve the physically unrealizable issue and lowers the dimension of the Poisson equation. Afterwards, we formally introduce the Lorenz '63 model that will serve as the testbed for the proposed filtering algorithm.

A. Particle Flow in the unstable subspace

From the calculation of the finite-time Lyapunov exponents and vectors introduced in Section III, suppose that the first n Lyapunov exponents are positive. Then, let $\mathbf{Q} \in \mathbb{R}^{m\times n}$ be the first n columns of the orthogonal matrix Q^*, which spans the unstable subspace of the tangent space.

Starting from the posterior states $\mathbf{X}^i_{t_k}$, they are propagated forward to time t_{k+1} by following (1). Then, we project each prior particle states $\mathbf{X}^i_{t^-_{k+1}}$ onto the unstable subspace through $\tilde{\mathbf{X}}^i_{t^-_{k+1}} = \mathbf{Q}^T\mathbf{X}^i_{t^-_{k+1}}$, which leaves the perpendicular components to be $\mathbf{X}^{i\perp}_{t^-_{k+1}} = \mathbf{X}^i_{t^-_{k+1}} - \mathbf{Q}\tilde{\mathbf{X}}^i_{t^-_{k+1}}$. Next, we project the observations at t_{k+1} to the unstable subspace. Assuming a linear sensor function $\mathbf{H} \in \mathbb{R}^{d\times m}$. Because the projection matrix $\mathbf{Q}$ is a mapping in the state space, we must start by mapping the observation into this space through a normalized projection factor $\hat{\mathbf{H}} = \mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}$. The projected observation in the state space is $\hat{\mathbf{Y}}_{t_{k+1}} = \hat{\mathbf{H}}\mathbf{Y}_{t_{k+1}}$, which is then projected onto the unstable space through $\tilde{\mathbf{Y}}_{t_{k+1}} = \mathbf{Q}^T\hat{\mathbf{Y}}_{t_{k+1}} = \mathbf{Q}^T\hat{\mathbf{H}}\mathbf{Y}_{t_{k+1}}$.

In this projected FPF method, only prior states in the UAS is updated, because after enough time, the state components in the stable direction will converge to the attractor. Upon completion of the particle flow within the pseudo-time, we have the projected posterior states $\tilde{\mathbf{X}}^i_{t_{k+1}}$, the perpendicular component of the prior states is brought back to obtain the posterior in the full state space $\mathbf{X}^i_{t_{k+1}} = \tilde{\mathbf{X}}^i_{t_{k+1}} + \mathbf{X}^{i\perp}_{t^-_{k+1}}$. The resulting projected FPF algorithm is shown in Algo. 1. Note that we use the Euler-discretization method for both the real-time t and the pseudo-time λ.

Algorithm 1 Projected FPF

INITIALIZATION
for $i = 1$ to N **do**
  Sample $\mathbf{X}^i_0$ from p_0, the initial distribution at $t = 0$
end for
$k = 0$
$t = 0$
ITERATION 1: $t \in [t_k, t_{k+1}]$
while $t \le t_k$ **do**
  for $i = 1$ to N **do**
    Propagate $\mathbf{X}^i_0$ to $\mathbf{X}^i_t$ following (1)
  end for
  $t = t + \Delta t$
end while
ITERATION 2: $t = t_{k+1}$
Calculate finite-time Lyapunov exponents in $[t_{k+1}, t_{k+2}]$.
Find the projection matrix $\mathbf{Q}$.
for $i = 1$ to N **do**
  Projection: $\tilde{\mathbf{X}}^i_t = \mathbf{Q}^T\mathbf{X}^i_t$
  Calculate: $\mathbf{X}^{i\perp}_t = \mathbf{X}^i_t - \mathbf{Q}\tilde{\mathbf{X}}^i_t$
end for
Calculate: $\hat{\mathbf{H}} = \mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}$
Projection :$\tilde{\mathbf{Y}}_t = \mathbf{Q}^T\hat{\mathbf{H}}\mathbf{Y}_t$.
$\lambda = 0$
for $i = 1$ to N **do**
  Set $S^i_t(0) = \tilde{\mathbf{X}}^i_t$
end for
while $\lambda \le 1$ **do**
  Calculate: $\hat{h}(\lambda) \approx \frac{1}{N}\sum_{i=1}^N h(S^i_t(\lambda))$
  Calculate $\tilde{\mathrm{K}}$ and $\tilde{\Omega}$ using (11).
  for $i = 1$ to N **do**
    Calculate $I^i_k(\lambda) = Y_k - \frac{1}{2}[h(S^i_k(\lambda)) + \hat{h}(\lambda)]$
    Update: $S^i_t(\lambda + \Delta\lambda) = S^i_t(\lambda) + [\tilde{\mathrm{K}}I^i_k(\lambda) + \frac{1}{2}\tilde{\Omega}]\Delta\lambda$
  end for
  $\lambda = \lambda + \Delta\lambda$
end while
for $i = 1$ to N **do**
  $\mathbf{X}^i_t = S^i_t(1) + \mathbf{X}^{i\perp}_t$
end for
$k = k + 1$
go to ITERATION 1

B. The Lorenz 1963 Model

The Lorenz 1963 model [18] is a simplified model of atmospheric convection. In particular, it models a 2-D fluid layer that is warmed from below and cooled from above. The model is given by:

$$d\begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} = \begin{bmatrix} -\sigma & \sigma & 0 \\ \rho & -1 & 0 \\ 0 & 0 & -\beta \end{bmatrix}\begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} dt + \begin{bmatrix} 0 \\ -x_t z_t \\ x_t y_t \end{bmatrix} dt, \tag{14}$$

where $\sigma = 10$, $\rho = 28$, and $\beta = \frac{8}{3}$. Under these choices of parameters, the system is chaotic with one positive Lya-

punov exponent being $\lambda = 0.9065$. This means that a small perturbation ϵ_0 grows exponentially as $\epsilon_t = \epsilon_0 e^{\lambda t}$. With this information, we can calculate the error doubling time to be $\tau_d = 0.76$ time units. The Lorenz '63 model has the well known butterfly-shape attractor in the x-y plane as shown in Fig. 1. Despite being a deterministic dynamical system, its behavior is non-periodic and appears random. For example, the saddle point, which separates the two wings of the butterfly, is an unstable equilibrium, meaning that trajectories diverge from it to either wing based on the initial conditions. This chaotic behavior poses many challenges to the filtering algorithm, as will be shown in next Section.

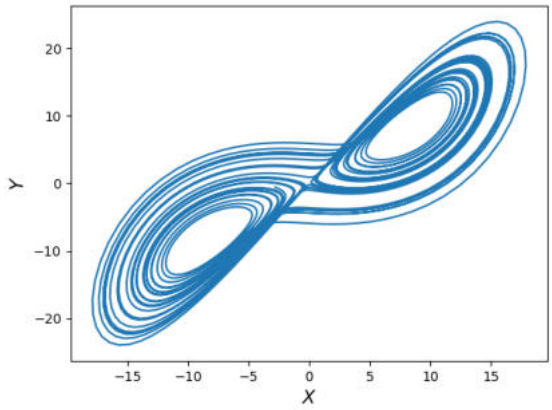


Fig. 1. The attractor for the Lorenz 1963 model in the x-y plane.

In this paper, we are interested in the stochastic variant of the Lorenz '63 model. Consider (14) in the abstract form of (1), then the stochastic Lorenz '63 model, which we simply referred as the Lorenz '63 model in the sequel, is written as:

$$d\mathbf{X}_t = \mathbf{b}(\mathbf{X}_t)dt + \boldsymbol{\sigma} d\mathbf{W}_t, \tag{15}$$

where $\mathbf{W}_t$ is a standard Brownian motion and $\boldsymbol{\sigma}\boldsymbol{\sigma}^T = \mathbf{I}$.

V. Numerical Results

In this section, we test the proposed projected particle flow method on the Lorenz '63 model. The linear sensor function that we use for data assimilation is an identity matrix, i.e. $\mathbf{H} = \mathbf{I} \in \mathbb{R}^3$, and the covariance matrix for the independent Gaussian random variables $\boldsymbol{\xi}$ in the observation process (1) is also an identity matrix, i.e., $\mathbf{R}_y = \mathbf{I} \in \mathbb{R}^3$. The observation is recorded every 0.4 time units, which is slightly more than half the error doubling time. The step size within the pseudo-time is $\Delta\lambda = 0.01$. An RK4 integrator is used for deterministic integration to compute the finite-time Lyapunov exponents and vectors, and an RK4-Maruyama scheme is used for stochastic integration, both using a step-size of 0.01. A total of 1000 Monte Carlo (MC) simulations are used to report the average results. The root-mean-square-error (RMSE) at time t is calculated as:

$$RMSE = \sqrt{\langle \mathbf{X}_t - \mathbb{E}[\mathbf{X}_t] \rangle}, \tag{16}$$

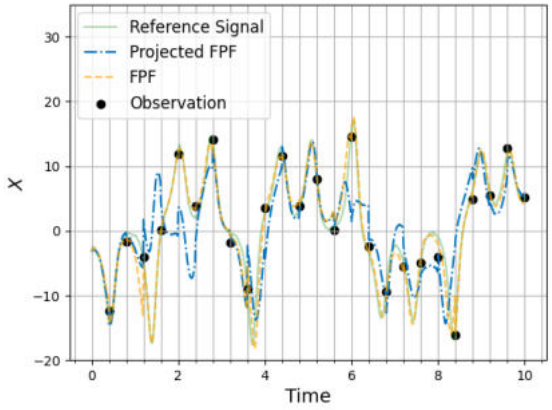
where $\mathbf{X}_t$ is the reference signal and $\langle \cdot \rangle$ represents the inner product.

There are three parameters that will affect the filtering performance: the number of particle N, the step size within the pseudo-time $\Delta\lambda$, and the basis function to numerically solve the Poisson equations in (8) and (9). In this paper, we assume a constant gain approximation to the Poisson equations as shown in (11) and investigate the other two parameters. Detailed analysis is given for $N = 5$ and $\Delta\lambda = 0.01$, then, we make comparisons by varying these two parameters.

We start by comparing filtering performances between the method in [11] (referred as the FPF method from here on) and the projected particle flow method proposed in this paper (referred as the projected FPF method from here on) with $N = 5$ and $\Delta\lambda = 0.01$. In Fig. 2, we compare the trajectories predicted by the FPF and projected FPF methods from a single MC run. Since states x and y have similar behavior, we only plot the trajectories of states x and z. It shows that the FPF method outperforms the projected FPF method, because its trajectory is closer to the reference signal. It is verified by the RMSE plot averaged over all three states shown in Fig. 3. Additionally, the average RMSE over 1000 MC runs for all three states is shown in Fig. 4. The grid over x-axis for both RMSE plots is given at every observation time.

The better performance from the FPF method is expected, because projecting the observation and prior states to a sub-space will lose information that may be valuable. However, the robustness of the projected FPF against the numerical singularity issue certainly out weights the loss of information. The results shown in Figs. 2 and 3 are from a specific MC run, in which the update within the pseudo-time is stable and no numerical singularity issue occurs. In Fig. 5, we show a specific MC run, in which numerical singularity occurs in the FPF method. Although the FPF performs better in the first six time units, it does not provide any information after the numerical singularity happens. In reality, over the total 1000 MC runs, more than half of them encounter the numerical singularity issue with the FPF method, while only around one tenth of them happen with the projected FPF method. With simulation time of 40 time units, the number of trials that encounters numerical singularity and the average RMSE over the entire time for both methods are shown in TABLE I.

One particular instance where numerical singularity happens in the FPF method is shown in Fig. 6. Each sub-figure shows an update step within the pseudo-time. The prior particles in Fig. 6 (a) are far from one another and the observation, additionally they are distributed in both "wings" of the Lorenz '63 attractor. They produce a large gain function K, with the L^2 norm being over 400. In contrast, the L^2 norm of K from the projected FPF method is 90. For 20 MC runs where the numerical singularity happen, the value of K within the pseudo-time for both methods is shown in Fig. 7, and it shows that the gain converges in the projectd FPF method but blows up in the FPF method. By inspecting the state update equation in (10), it is a first-order optimization method with minimum at $I_k^i(\lambda) = 0$ (with the sensor function being $\mathbf{H} = \mathbf{I}$, we



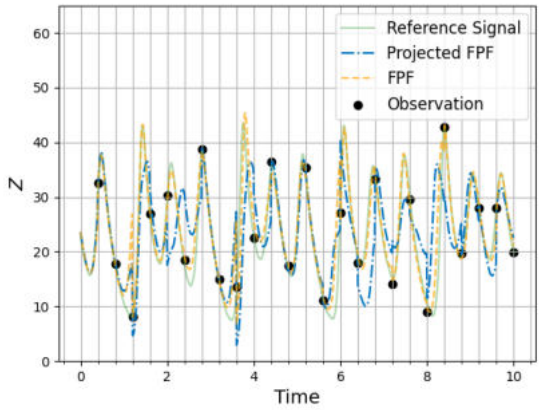


Fig. 2. Comparison of the predicted trajectories from the FPF algorithm and the projected FPF method. The reference signal is shown in green, and observations are shown in black dots.

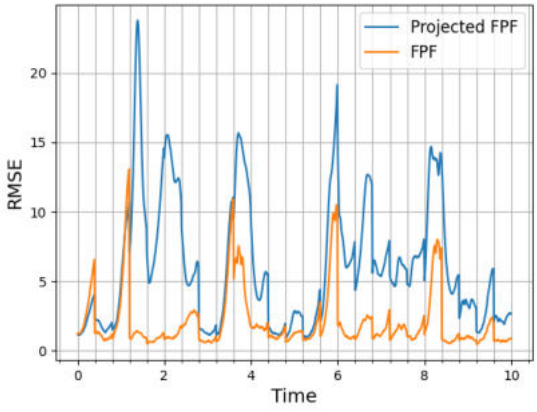


Fig. 3. Comparison of the RMSE averaged over three states of the FPF and the projected FPF methods from the same MC run as in Fig. 2.

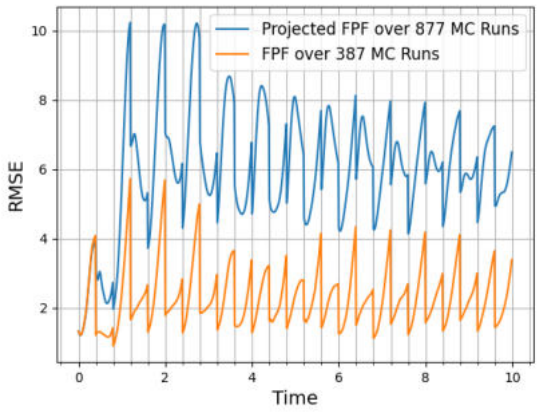


Fig. 4. Comparison of the RMSE averaged over three states and 1000 MC runs of the FPF and the projected FPF methods.

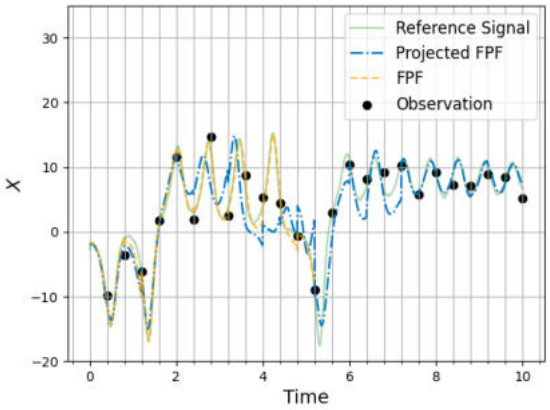
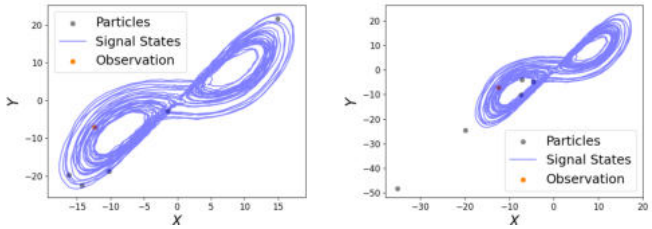


Fig. 5. Comparison of the FPF and projected FPF methods, where numerical singularity happens for the FPF method.

have $\Omega = 0$ in (10)). Thus, a large gain function K tends to overshoot the extremum and cause unstable pseudo-time update as shown in Fig. 6 (b)-(d). Fig. 8 depicts the average states of the x and y components within the pseudo-time from 5 MC runs where numerical singularity occurs. It shows the growing oscillatory behavior, which corresponds to an unstable feedback loop.

Last but not least, we test both methods with different choices of particle number N and step size $\Delta\lambda$, and we report the number of trials with numerical singularity issue, the RMSE, and the runtime over 1000 MC trials in Table I. Although the runtime for the projected FPF is always higher than the original FPF method due to the computation of the fundamental matrix and the associated linear algebra for the projection operations, it is likely worth it as the runtime increases significantly with more particle or finer step size in order to mitigate the numerical singularity issue. Table I also shows that the numerical singularity issue can be solved with small enough step size, but increasing the number of particles



(a) $\lambda = 0$ (b) $\lambda = 0.01$

(c) $\lambda = 0.02$ (d) $\lambda = 0.03$

Fig. 6. Illustration of numerical singularity within the pseudo-time.

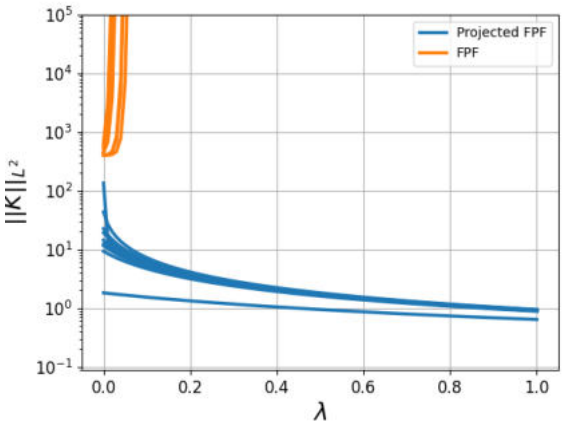


Fig. 7. The norm of the gain function K within the pseudo-time when the FPF method has a numerical singularity.

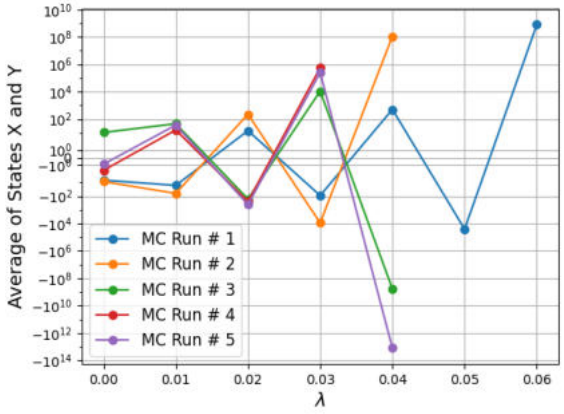


Fig. 8. Average of x and y states within the pseudo-time for 5 MC runs where numerical singularity occurs.

does not yield the same discernible resolution.

TABLE I
NUMBER OF TRIALS WITH NUMERICAL SINGULARITY, RMSE, AND RUNTIME OVER 1000 MC TRIALS FOR THE FPF AND PROJECTED FPF METHODS

		Projected FPF	FPF
$N = 5$, $\Delta\lambda = 0.01$	# Singularity	123	613
	RMSE	6.07	2.24
	Runtime (s)	1457	862
$N = 80$, $\Delta\lambda = 0.01$	# Singularity	37	344
	RMSE	6.05	1.92
	Runtime (s)	17265	13014
$N = 5$, $\Delta\lambda = 0.001$	# Singularity	0	0
	RMSE	6.05	1.93
	Runtime (s)	8560	7008

VI. CONCLUSIONS

In this paper, we proposed the novel projected FPF algorithm, aiming to mitigate the physically unrealizable and numerical singularity issues that occur when applying the original FPF method to the nonlinear chaotic dynamical system, in particular the Lorenz '63 model. The new method bringing in the local dynamical information through projecting observations and states onto the unstable subspace computed from the Lyapunov exponents and vectors. When applying to high-order dynamical systems in the future, such as the Lorenz '96 model, the number of particles become critical. Thus, we intentionally pick only 5 particles in this work when testing on the low-order Lorenz '63 model. We discovered that more than half of the 1000 MC runs failed when using the original FPF method, while the projection method effectively mitigate this issue.

In the future, we would like to apply this method to other higher-order dynamical systems. The reduction in dimensions of the projected subspace will become more apparent, and we will test more advanced numerical methods in solving the Poisson equation. For example, the piecewise constant approximation and other basis functions in the Galerkin method. We will also explore in more detail about the pseudo-update on the unstable subspace, in particular the reason why local dynamical information can help with this numerical singularity issue, and how we can further improve this idea to reduce the RMSE.

REFERENCES

[1] J.-I. Yano, M. Z. Ziemiański, M. Cullen, P. Termonia, J. Onvlee, L. Bengtsson, A. Carrassi, R. Davy, A. Deluca, S. L. Gray *et al.*, "Scientific challenges of convective-scale numerical weather prediction," *Bulletin of the American Meteorological Society*, vol. 99, no. 4, pp. 699–710, 2018.

[2] T. Bengtsson, P. Bickel, and B. Li, "Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems," in *Probability and statistics: Essays in honor of David A. Freedman*. Institute of Mathematical Statistics, 2008, vol. 2, pp. 316–335.

[3] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson, "Obstacles to high-dimensional particle filtering," *Monthly Weather Review*, vol. 136, no. 12, pp. 4629–4640, 2008.

[4] P. J. Van Leeuwen, "Nonlinear data assimilation in geosciences: an extremely efficient particle filter," *Quarterly Journal of the Royal Meteorological Society*, vol. 136, no. 653, pp. 1991–1999, 2010.

[5] H. C. Yeong, R. T. Beeson, N. S. Namachchivaya, and N. Perkowski, "Particle filters with nudging in multiscale chaotic systems: With application to the lorenz'96 atmospheric model," *Journal of Nonlinear Science*, vol. 30, pp. 1519–1552, 2020.

[6] F. Daum and J. Huang, "Nonlinear filters with log-homotopy," in *Signal and Data Processing of Small Targets 2007*, vol. 6699. SPIE, 2007, pp. 423–437.

[7] ——, "Particle flow for nonlinear filters with log-homotopy," in *Signal and Data Processing of Small Targets 2008*, vol. 6969. SPIE, 2008, pp. 414–425.

[8] ——, "Nonlinear filters with particle flow," in *Signal and Data Processing of Small Targets 2009*, vol. 7445. SPIE, 2009, pp. 315–323.

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

[10] T. Yang, P. G. Mehta, and S. P. Meyn, "Feedback particle filter," *IEEE transactions on Automatic control*, vol. 58, no. 10, pp. 2465–2480, 2013.

[11] T. Yang, H. A. Blom, and P. G. Mehta, "The continuous-discrete time feedback particle filter," in *2014 American Control Conference*. IEEE, 2014, pp. 648–653.

[12] T. Yang, R. S. Laugesen, P. G. Mehta, and S. P. Meyn, "Multivariable feedback particle filter," *Automatica*, vol. 71, pp. 10–23, 2016.

[13] R. S. Laugesen, P. G. Mehta, S. P. Meyn, and M. Raginsky, "Poisson's equation in nonlinear filtering," *SIAM Journal on Control and Optimization*, vol. 53, no. 1, pp. 501–525, 2015.

[14] A. Taghvaei, P. G. Mehta, and S. P. Meyn, "Diffusion map-based algorithm for gain function approximation in the feedback particle filter," *SIAM/ASA Journal on Uncertainty Quantification*, vol. 8, no. 3, pp. 1090–1117, 2020.

[15] R. Beeson and N. Sri Namachchivaya, "Particle filtering for chaotic dynamical systems using future right-singular vectors," *Nonlinear Dynamics*, vol. 102, no. 2, pp. 679–696, 2020.

[16] A. Trevisan, M. D'Isidoro, and O. Talagrand, "Four-dimensional variational assimilation in the unstable subspace and the optimal subspace dimension," *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, vol. 136, no. 647, pp. 487–496, 2010.

[17] L. Palatella, A. Carrassi, and A. Trevisan, "Lyapunov vectors and assimilation in the unstable subspace: theory and applications," *Journal of Physics A: Mathematical and Theoretical*, vol. 46, no. 25, p. 254020, 2013.

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

[19] V. I. Oseledets, "A multiplicative ergodic theorem. characteristic ljapunov, exponents of dynamical systems," *Trudy Moskovskogo Matematicheskogo Obshchestva*, vol. 19, pp. 179–210, 1968.

[20] G. Benettin, L. Galgani, A. Giorgilli, and J.-M. Strelcyn, "Lyapunov characteristic exponents for smooth dynamical systems and for hamiltonian systems; a method for computing all of them. part 1: Theory," *Meccanica*, vol. 15, pp. 9–20, 1980.