A Drift Homotopy Implicit Particle Filter Method for Nonlinear Filtering problems

Xin Li * Feng Bao† Kyle Gallivan ‡

Abstract

In this paper, we develop a drift homotopy implicit particle filter method. The methodology of our approach is to adopt the concept of drift homotopy in the resampling procedure of the particle filter method for solving the nonlinear filtering problem, and we introduce an implicit particle filter method to improve the efficiency of the drift homotopy resampling procedure. Numerical experiments are carried out to demonstrate the effectiveness and efficiency of our drift homotopy implicit particle filter.

Keywords: Nonlinear filtering problem, particle filter, drift homotopy dynamics, implicit sampling

Mathematics subject classifications: 60G35, 62M20, 93E11

1 Introduction

The goal of nonlinear filtering problems is to make the best estimation for the state of some stochastic dynamical system based on its partial noisy observations. As a key mathematical tool for data assimilation, nonlinear filtering methods have various applications in many scientific and engineering areas, such as weather forecasting, parameter estimation, signal processing, target tracking, and machine learning [34].

There are two types of approaches to solve the nonlinear filtering problem. The first type formulates the conditional distribution of the state of the target dynamical system by stochastic partial (or ordinary) differential equations, and then computes approximated distributions through numerical solutions of the equations [5–13, 16, 21, 26, 35]. The other type of approach is called the "Bayesian filter", in which Bayesian inference is used to incorporate observational data into the dynamical model to estimate the state. In this work, we focus on the Bayesian approach due to its wide acceptance by practitioners. There are two categories of Bayesian filter: Kalman-type filters and particle filters. The main idea of Kalman-type filers is to use the classic Kalman-Bucy filter, which can solve the linear filtering problem analytically, to solve a linearized filtering

*Department of Mathematics, Florida State University, Tallahassee, Florida.
†Department of Mathematics, Florida State University, Tallahassee, Florida, (fbao@fsu.edu).
‡Department of Mathematics, Florida State University, Tallahassee, Florida, (kgallivan@fsu.edu).

problem. Well-known Kalman-type filters include the ensemble Kalman filter, the extended Kalman filter, and the unscented Kalman filter [19, 20, 23, 31, 32]. Although Kalman filters are easy to implement and they are efficient in solving high dimensional problems, the major drawback of the Kalman type filters is their poor performance when the dynamical system and the observation function are highly nonlinear. The linearized problem does not provide a good approximation to the original nonlinear filtering problem, and the quality of the state estimate degrades significantly. The particle filter method (also called the sequential Monte Carlo method), on the other hand, is designed specifically to solve nonlinear filtering problems. In contrast to the Kalman filter framework, which propagates and updates Gaussian distributions, the particle filter uses a set of weighted random samples (particles) to describe arbitrary conditional distributions of the target state [3,4,15,17,22,25]. The weights on the particles are used to incorporate observational information, and they are assigned by Bayesian inference. Through flexible Monte Carlo sampling, a particle filter can effectively handle nonlinear dynamics and nonlinear observations. The primary challenge of particle filter is that the errors caused by Monte Carlo sampling can accumulate through the sequential sampling procedures. Therefore, particle filters often suffer from the so-called "degeneracy" problem. In other words, after several estimation steps, most particles tend to lie in insignificant regions of the distribution, hence the effective ensemble size is dramatically reduced [30].

One of the most effective approaches to address the degeneracy problem in the particle filter is "resampling". The goal of resampling is to rejuvenate the particle cloud and relocate particles from low density regions to high density regions. Usually, a resampling step is implemented after (or combined with) Bayesian inference and it generates a set of particles (or moves current particles) that follow the desired conditional distribution of the target state. Successful resampling methods include sequential importance sampling with resampling (the benchmark method), the auxiliary particle filters, the Markov Chain Monte Carlo particle filter, the drift homotopy particle filter, and the implicit particle filter [1,2,14,22,27,29,33]. While all of these methods can mitigate the degeneracy problem to some extent, they all have their disadvantages and drawbacks.

In this paper, we develop a drift homotopy implicit particle filter method that combines the drift homotopy particle filter [24, 27] and the implicit particle filter [14]. The central concept of the drift homotopy particle filter is to construct a sequence of intermediate systems called drift homotopy dynamics, and then transport particles by using the Markov Chain Monte Carlo (MCMC) sampling method, which is driven by those intermediate homotopy systems, to high density regions of the desired state distribution. The drift homotopy dynamics are usually designed in a way so that the observational data play a more important role in the first few particle transportation steps, and then the original filtering dynamical model is incorporated into the state distribution gradually. In this way, the drift homotopy particle filter is different from most traditional Bayesian approaches, which simulate dynamical models first and then incorporate data through Bayesian inference. As a result, the drift homotopy systems provide a mechanism to "trust the observational data" first, which typically results more robust estimation performance. The main drawback of the drift homotopy particle filter is that the MCMC sampling procedure is time consuming – especially carried out repeatedly through the sequence of drift homotopy dynamics , and thus the drift homotopy particle filter is

not an efficient method. The primary effort of the implicit particle filter is to carry out an implicit sampling procedure, which works by first picking target probabilities and then looking for particles that assume them, so that the particles are guided to the high probability region. In practical implementation, the implicit sampling procedure is achieved by optimization, and the efficiency of implicit sampling is based on the shape of state distribution, which is governed by the state dynamics and the observational data. Therefore, when the state dynamical model and the observational data do not align well, the optimization task in the implicit particle filter could be very challenging.

Our motivation for developing a drift homotopy implicit particle filter (DHIPF) method is to exploit the advantages of the drift homotopy particle filter and the implicit particle filter while alleviating their disadvantages. Specifically, we shall adopt the general drift homotopy framework and utilize a sequence of drift homotopy dynamics to transport particles. However, instead of using the MCMC sampling method to move particles slowly, we treat the sampling procedure for intermediate drift homotopy systems as a sequence of nonlinear filtering problems and then use the implicit particle filter to solve those filtering problems. Since the implicit sampling in the implicit particle filter is achieved by optimization, the implicit particle filter sampler for intermediate drift homotopy systems is much more efficient than the MCMC sampling method. In this connection, the application of the implicit particle filter in DHIPF can significantly improve efficiency of the conventional drift homotopy particle filter. On the other hand, since the observational data play a more important role in the first few particle transportation steps in the drift homotopy systems, our DHIPF method could endow the implicit particle filter with the mechanism of "trust observational data first", which can make the DHIPF obtain more robust estimation results.

The rest of this paper is organized as follows. In Section 2, we introduce the general mathematical formulation of the nonlinear filtering problem. In Section 3, we introduce the state-of-the-art approach to solve the filtering problem, i.e. the particle filter, and briefly discuss the drift homotopy particle filter and the implicit particle filter. Then, in Section 4, we combine the drift homotopy particle filter and the implicit particle filter to establish our drift homotopy implicit particle filter method. Numerical experiments that illustrate the performance of our method are given in Section 5.

2 The nonlinear filtering problem

An optimal filtering problem is usually described by a system of stochastic differential equations (SDEs)

$$\begin{aligned} dX_t =& f(X_t)dt + \sigma_t dW_t, && \text{(State)} \\ dY_t =& g(X_t)dt + dV_t. && \text{(Observation)} \end{aligned} \tag{1}$$

The first equation in (1) is a state equation that models the state of a dynamical system, where W_t is a standard Brownian motion, $f(X_t)dt$ is the drift term and the $\sigma_t dW_t$ integral is called the diffusion term. The second equation is the observational equation that gives partial noisy observations of X_t. In practical applications, this continuous version of the optimal filtering problem is often discretized and represented

by the following discrete state-space model

$$\begin{aligned} X_{n+1} =& f(X_n) + \sigma_n w_n, \qquad \text{(State)} \\ Y_{n+1} =& g(X_{n+1}) + v_n, \quad \text{(Observation)} \end{aligned} \tag{2}$$

where we have incorporated the temporal discretization steps Δt into the model. In this way, the sequence $\{X_n\}_n \in \mathbb{R}^d$ describes the state of the stochastic dynamical system, the function $f : \mathbb{R}^d \to \mathbb{R}^d$ now plays the role of the drift term in the continuous state equation in (1), and the state of X is perturbed by a sequence of d-dimensional standard Gaussian noises $\{w_n\}_n$ with their coefficients $\{\sigma_n\}_n$, and $Y_{n+1} \in \mathbb{R}^m$ is the m-dimensional partial noisy measurement on X_{n+1} through the observation function $g : \mathbb{R}^d \to \mathbb{R}^m$, which is also perturbed by a Gaussian noise v_n independent from w_n with the standard deviation R. For the discretized optimal filtering problem (2), the first stochastic process in (2) is called the "state process" and the second process is called the "observation process". When the functions f and g are nonlinear functions, the filtering problem is called the "nonlinear filtering problem". The goal of the nonlinear filtering problem is to find the best estimate for $\Phi(X_{n+1})$ given the observational data $Y_{1:n+1} := \{Y_1, Y_2, \cdots, Y_{n+1}\}$, where Φ is a test function that represents the quantity of interest in the nonlinear filtering problem. Mathematically, we aim to find the "optimal filter" $\tilde{\Phi}(X_{n+1})$ as a conditional expectation given $Y_{1:n+1}$, i.e.

$$\tilde{\Phi}(X_{n+1}) := \mathbb{E}[\Phi(X_{n+1})|Y_{1:n+1}].$$

The standard approach to solve the optimal filtering problem (2) is the "Bayesian filter", which aims to find the best estimate for the conditional probability density function (pdf) $p(X_{n+1}|Y_{1:n+1})$ of the state through recursive Bayesian inference. Then, the conditional pdf $p(X_{n+1}|Y_{1:n+1})$, which is also called the "filtering density", can be used to calculate the optimal filter $\tilde{\Phi}$. Specifically, the Bayesian filter is composed of two steps: a prediction step and an update step. For the given conditional pdf $p(X_n|Y_{1:n})$ at the time instant n, the prediction step is carried out by the following Chapman-Kolmogorov formula,

$$p(X_{n+1}|Y_{1:n}) = \int p_f(X_{n+1}|X_n) p(X_n|Y_{1:n}) dX_n, \qquad \text{(Prediction)} \tag{3}$$

where $p_f(X_{n+1}|X_n)$ is the transition probability associated with the state dynamical function f, and $p(X_{n+1}|Y_{1:n})$ is called the prior pdf, which predicts the state of X_{n+1} before receiving the new observational data. In the update step, we apply the Bayes formula to incorporate the observational data Y_{n+1} to update the prior pdf and get a posterior distribution $p(X_{n+1}|Y_{1:n+1})$ as following

$$p(X_{n+1}|Y_{1:n+1}) = \frac{p(Y_{n+1}|X_{n+1}) p(X_{n+1}|Y_{1:n})}{p(Y_{n+1}|Y_{1:n})}, \qquad \text{(Update)} \tag{4}$$

where $p(Y_{n+1}|X_{n+1})$ is the likelihood function that measures the discrepancy between the state and the observation, and the denominator in (4) is given by

$$p(Y_{n+1}|Y_{1:n}) = \int p(Y_{n+1}|X_{n+1}) p(X_{n+1}|Y_{1:n}) dX_{n+1},$$

which normalizes the posterior $p(X_{n+1}|Y_{1:n+1})$.

In this work, we aim to develop an efficient and effective particle filter method to implement the Bayesian filter's "Prediction-Update" framework.

3 The particle filter approach

3.1 The generic particle filter framework

The main strategy of the particle filter is to use a cloud of samples, which are called particles, to represent conditional distributions, and use the recursive Bayesian filter framework (3) - (4) to propagate and updates the particle cloud. In what follows, we give a brief discussion to introduce the general framework of particle filters.

At the time instant n, assume that we have a set of N_p particles that form an empirical distribution $\tilde{p}(X_n|Y_{1:n})$, which approximates the conditional pdf $p(X_n|Y_{1:n})$. We denote this set of particles by $\{x_n^{(i)}\}_{i=1}^{N_p}$, and the empirical distribution is defined as

$$\tilde{p}(X_n|Y_{1:n}) := \frac{1}{N_p}\sum_{i=1}^{N_p}\delta_{x_n^{(i)}}(X_n), \tag{5}$$

where δ_x is the Dirac delta function. In the prediction step, we propagate each sample $x_n^{(i)}$ in the particle cloud through the state dynamical model f to get a predicted sample $\tilde{x}_{n+1}^{(i)}$. In this way, the ensemble of predicted particles $\{\tilde{x}_{n+1}^{(i)}\}_{i=1}^{N_p}$ form an empirical distribution $\tilde{\pi}(X_{n+1}|Y_{1:n})$ defined as

$$\tilde{\pi}(X_{n+1}|Y_{1:n}) := \frac{1}{N_p}\sum_{i=1}^{N_p}\delta_{\tilde{x}_{n+1}^{(i)}}(X_{n+1}), \tag{6}$$

which is an approximation for the prior distribution $p(X_{n+1}|Y_{1:n})$.

In the update step, after receiving the new measurement Y_{n+1}, we carry out Bayesian inference through the Bayes formula (4) to incorporate the new observational data to get an approximation for the posterior distribution. Specifically, we use the empirical distribution $\tilde{\pi}(X_{n+1}|Y_{1:n})$ to replace the prior distribution $p(X_{n+1}|Y_{1:n})$ in (4), and obtain

$$\tilde{\pi}(X_{n+1}|Y_{1:n+1}) = \frac{p(Y_{n+1}|X_{n+1})\tilde{\pi}(X_{n+1}|Y_{1:n})}{\int p(Y_{n+1|X_{n+1}})\tilde{\pi}(X_{n+1}|Y_{1:n})dX_{n+1}}, \tag{7}$$

where the likelihood function $p(Y_{n+1}|X_{n+1})$ of the Gaussian noise v with the standard deviation R is given by $p(Y_{n+1}|X_{n+1}) = \frac{1}{\sqrt{(2\pi R^2)^m}}\exp\Big(-\frac{\|g(X_{n+1})-Y_{n+1}\|^2}{2R^2}\Big)$. With the empirical distribution $\tilde{\pi}(X_{n+1}|Y_{1:n})$ defined in (6), the Bayesian inference formula (7) can be implemented by the following update scheme

$$\tilde{\pi}(X_{n+1}|Y_{1:n+1}) = \frac{\sum_{i=1}^{N_p} p(Y_{n+1}|\tilde{x}_{n+1}^{(i)})\delta_{\tilde{x}_{n+1}^{(i)}}(X_{n+1})}{\sum_{i=1}^{N_p} p(Y_{n+1}|\tilde{x}_{n+1}^{(i)})}. \tag{8}$$

We let $\alpha_{n+1}^{(i)} \propto p(Y_{n+1}|\tilde{x}_{n+1}^{(i)})$ be the importance weight corresponding to the particle $\tilde{x}_{n+1}^{(i)}$ such that $\sum_{i=1}^{N_p} \alpha_{n+1}^{(i)} = 1$. Then, the empirical distribution $\tilde{\pi}(X_{n+1}|Y_{1:n+1})$ is an approximation for the posterior distribution $p(X_{n+1}|Y_{1:n+1})$, where we have

$$\tilde{\pi}(X_{n+1}|Y_{1:n+1}) = \sum_{i=1}^{N_p} \alpha_{n+1}^{(i)} \delta_{\tilde{x}_{n+1}^{(i)}}(X_{n+1}).$$

In practice, due to the extra uncertainties involved in the observational data and the sequential sampling errors, the weights on many particles tend to be negligible after several recursive steps, and only a few particles have very large weights, which significantly reduces the effective ensemble size. This fact of losing the effectiveness of particle sizes is often called the "degeneracy" of particles. To address the degeneracy problem, a *resampling step* is introduced to re-generate the particle cloud with equally weighted particles that describe the empirical distribution $\tilde{\pi}(X_{n+1}|Y_{1:n+1})$. In the benchmark bootstrap particle filter, which is also known as the sequential importance sampling with resampling method, people use importance sampling to generate N_p samples, denoted by $\{x_{n+1}^{(i)}\}_{i=1}^{N_p}$, which include more copies of particles in the weighted particle cloud $\{\tilde{x}_{n+1}^{(i)}\}_{i=1}^{N_p}$ and discard the low weight ones. In this way, the resampled particles give us the following empirical distribution

$$\tilde{p}(X_{n+1}|Y_{1:n+1}) := \frac{1}{N_p} \sum_{i=1}^{N_p} \delta_{x_{n+1}^{(i)}}(X_{n+1}).$$

Although the importance sampling in the bootstrap particle filter is a successful strategy to increase the effective particle size in recursion, simply reproducing more particles in the original predicted particle set $\{\tilde{x}_{n+1}^{(i)}\}_{i=1}^{N_p}$ still suffers from the degeneracy problem since the prediction step is not informed by the observational information and the predicted particles may not provide good candidate particles to be reproduced. To conquer the degeneracy of particles, several data informed resampling methods have been developed, such as the auxiliary transportation, the Markov Chain Monte Carlo resampling, the drift homotopy particle filter, the implicit particle filter. In this work, we combine the drift homotopy particle filter and the implicit particle filter to develop an efficient drift homotopy implicit particle filter method that takes the advantages of both methods. To proceed, we shall introduce the drift homotopy particle filter and the implicit particle filter in the following.

3.2 The drift homotopy particle filter

Homotopy is a mathematical process that continuously transforms one function to another. When adopting homotopy in the particle filter, we design a homotopy process that transforms the drift term in the state dynamics gradually from an intermediate drift homotopy system, so that the sampling of the desired filtering density can be implemented effectively.

In this connection, the key of the drift homotopy particle filter is to construct a sequence of stochastic dynamical systems with modified drift terms that interpolate

between the original and modified drifts in the state model. To proceed, assume that we have an empirical distribution that describes the conditional pdf $p(X_n|Y_{1:n})$ at the time instant n with equally weighted particles $\{x_n^{(i)}\}_{i=1}^{N_p}$. For a given particle $x_n^{(i)}$, the "ideal" particle at the time instant $n+1$ that could be used to describe the posterior distribution $p(X_{n+1}|Y_{1:n+1})$ should follow the distribution $p(x_{n+1}|x_n^{(i)}, Y_{1:n+1})$, which is defined by

$$p(X_{n+1}|x_n^{(i)}, Y_{1:n+1}) \propto p_f(X_{n+1}|x_n^{(i)})p(Y_{n+1}|x_{n+1}), \tag{9}$$

where $p_f(X_{n+1}|x_n^{(i)})$ is the transition probability that describes the conditional distribution of X_{n+1} with the given initial state at the time instant n as $X_n = x_n^{(i)}$, and the likelihood $p(Y_{n+1}|X_{n+1})$ incorporates the observational data Y_{n+1} into the conditional distribution $p(X_{n+1}|x_n^{(i)}, Y_{1:n+1})$. In order to generate a sample that follows the posterior distribution, which considers the new observational data $Y_{1:n+1}$, instead of using the importance sampling method which only reproduces more copies of high density propagation samples, one could use the Markov Chain Monte Carlo (MCMC) sampling method to generate a sample $x_{n+1}^{(i)}$ that follows the conditional distribution $p(X_{n+1}|x_n^{(i)}, Y_{1:n+1})$. However, it is well known that the effectiveness and efficiency of MCMC sampling depends on the complexity of the target distribution. When the dynamics of the state process is complicated, it is difficult for the MCMC method to generate the sample $x_{n+1}^{(i)}$. Moreover, in the case that the practical observational data has large deviation from the prediction, the resulting posterior distribution might have large covariance, which makes the MCMC procedure have even worse performance.

The primary contribution of the drift homotopy particle filter is to improve the performance of the MCMC procedure to effectively generate a sample that follows the posterior distribution. Also, it could provide a mechanism that allows us to "trust the data" first. In the drift homotopy particle filter, instead of generating a sample from $p(X_{n+1}|x_n^{(i)}, Y_{1:n+1})$ by using MCMC sampling directly, we introduce a sequence of drift homotopy dynamical systems given as follows

$$X^* = (1-\beta_l)b(X) + \beta_l f(X) + \sigma_n w_n, \qquad l = 0, 1, 2, \cdots, L \tag{10}$$

where the function b is artificially defined intermediate drift term, which is different from the original drift f, and $\{\beta_l\}_{l=0}^{L}$ is a constant sequence increasing from 0 to 1. Therefore, when $l = 0$, we have $\beta_0 = 0$, and the dynamical system (10) only contains the intermediate function b. On the other hand, when $l = L$, i.e. $\beta_L = 1$, the intermediate drift term b is gone and (10) coincides the original state dynamical model in the nonlinear filtering problem (2). For a specific drift homotopy level l, $0 \le l \le L-1$, the dynamical system (10) is driven by the combined model $(1-\beta_l)b(X^l) + \beta_l f(X^l)$. To generate a sample that follows the desired conditional distribution $p(X_{n+1}|x_n^{(i)}, Y_{1:n+1})$ (as described in (9)), we incorporate the observational data Y_{n+1} through the likelihood function of the dynamical system (10). Since the target distribution $p(X_{n+1}|x_n^{(i)}, Y_{1:n+1})$ is conditioned on $x_n^{(i)}$, we take the state X of the drift homotopy dynamics (10) to be the particle $x_n^{(i)}$, i.e. $X = x_n^{(i)}$, which is a sample in the previous conditional particle cloud $\{x_n^{(i)}\}_{i=1}^{N_p}$. As a result, we obtain a sequence of drift homotopy distributions $\{p_l(X^*|x_n^{(i)}, Y_{1:n+1})\}_{l=1}^{L}$

for each particle $x_n^{(i)}$ through the following proportional relation

$$p_l(X^*|x_n^{(i)}, Y_{1:n+1}) \propto p_l(X^*|x_n^{(i)})p(Y_{n+1}|X^*), \tag{11}$$

where $p_l(X^*|x_n^{(i)})$ is the transition probability of the system (10) determined by the combined dynamics. In this way, the distribution of the L-th drift homotopy step is the desired posterior distribution given that $p_L(X^*|x_n^{(i)}) = p_f(X_{n+1}|x_n^{(i)})$.

To carry out the drift homotopy particle filter, we first choose $x_n^{(i)}$ as the initial state of X and use the MCMC method to generate a sample, denoted by $x_{n+1}^{*,0}$, from the first homotopy distribution $p_0(X^*|x_n^{(i)}, Y_{1:n+1})$. For the l-th drift homotopy level, $l = 0, 1, 2, \cdots, L-1$, assume that we have the sample $x_{n+1}^{*,l}$, we let $x_{n+1}^{*,l+1}$ be the initial state of the drift homotopy system (10) and use the MCMC method to generate a sample, denoted by $x_{n+1}^{*,l+1}$ from the homotopy distribution $p_{l+1}(X^*|x_n^{(i)}, Y_{1:n+1})$. As a result, the sample $x_{n+1}^{*,L}$ that we generate in the iterative drift homotopy procedures gives a sample that follows the desired posterior distribution $p(X_{n+1}|x_n^{(i)}, Y_{1:n+1})$, and we let $x_{n+1}^{(i)} = x_{n+1}^{*,L}$.

From the above discussion, we can see that the main theme of the drift homotopy particle filter is to transport the particle $x_n^{(i)}$ gradually to $x_{n+1}^{(i)}$, which is then used to formulate the posterior distribution. In order to fully assimilate the observational information into the posterior distribution, we choose the intermediate drift term b so that the initial transition probability $p_0(X^*|x_n^{(i)})$ is well-aligned with the likelihood function $p(Y_{n+1}|X^*)$. Then, as the drift homotopy distributions $\{p_l(X^*|x_n^{(i)}, Y_{1:n+1})\}_l$ morph gradually to the ultimate posterior distribution $p(X_{n+1}|x_n^{(i)}, Y_{1:n+1})$, the original state dynamics f is incorporated. In this way, the homotopy procedure builds a bridge to connect the sample $x_n^{(i)}$ in the previous particle cloud at the time instant n to the particle $x_{n+1}^{(i)}$ for the desired posterior distribution. Moreover, note that the combined dynamical system only contains the intermediate drift term b in the first drift homotopy step, which is designed to be well-aligned with the observational data. Therefore, the observational data would have more influence in the drift homotopy distribution. In this way, the drift homotopy particle filter also creates a mechanism that allows us to "trust the data first".

3.3 The implicit particle filter

The central concept of the implicit particle filter is to update the particles by first deriving implicit transportation probabilities, which construct the conditional distribution of the target, and then determine particle locations that assume them. As a result, the implicit sampling procedure guides the particles one by one to the high probability domain of the desired posterior distribution, and therefore it can effectively address the degeneracy problem of the particle filter approach.

In the standard particle filter, assume that we have the particle $x_n^{(i)}$ at the time step n, then the predicted particle location $\tilde{x}_{n+1}^{(i)}$ is propagated from the particle $x_n^{(i)}$ through the state dynamics introduced in (2). According to the Bayesian update scheme (4),

the weight on $\tilde{x}_{n+1}^{(i)}$ is given by

$$\alpha_{n+1}^{(i)} = \frac{p(\tilde{x}_{n+1}^{(i)}|x_n^{(i)})p(Y_{n+1}|\tilde{x}_{n+1}^{(i)})}{C},$$

where C is the normalization constant. The set of weights $\{\alpha_{n+1}^{(i)}\}_{i=1}^{N_p}$, together with the predicted particle locations $\{\tilde{x}_{n+1}^{(i)}\}_{i=1}^{N_p}$, now describe the desired posterior density. In the implicit particle filter method, instead of simply reproducing more copies of particles in the predicted particle cloud with high weights, an implicit sampling procedure is used to generate "optimal" particle locations, i.e. $\{\hat{x}_{n+1}^{(i)}\}_{i=1}^{N_p}$, that follow the posterior distribution directly.

To achieve this goal, we first choose a reference random variable ξ with a predetermined pdf that is easy to sample. Then, we treat the desired optimal sample variable $\hat{x}_{n+1}^{(i)} := \psi^{n+1,i}(\xi)$ as a function of ξ, which is indexed by both the time instant and the specific choice of particle. The purpose of the function ψ is to connect highly probable values of ξ to highly probable values of $\hat{x}_{n+1}^{(i)}$, which follow the posterior distribution. To obtain the mapping ψ, we define a function $F^{n+1,i}$ corresponding to each time instant $n+1$ and each particle i, such that

$$\exp\Big(-F^{n+1,i}(\hat{x}_{n+1}^{(i)})\Big) := p(\hat{x}_{n+1}^{(i)}|x_n^{(i)})p(Y_{n+1}|\hat{x}_{n+1}^{(i)}), \tag{12}$$

and solve the equation

$$F^{n+1,i}\Big(\psi^{n+1,i}(\xi)\Big) - \gamma^{n+1,i} = \frac{\xi^T\xi}{2} \tag{13}$$

to get the function $\psi^{n+1,i}$. The random variable $\gamma^{n+1,i}$ in the above equation is an additive factor introduced to make the above equation solvable, and it is typical choose $\gamma^{n+1,i} = \min F^{n+1,i} + \lambda$, where λ represents a small perturbation. When implementing the implicit particle filter numerically, optimization type numerical solvers are needed to calculate $\psi^{n+1,i}$ through (13).

Once the function $\psi^{n+1,i}$ is determined, we can obtain the particle set $\{\hat{x}_{n+1}^{(i)}\}_{i=1}^{N_p}$ since the position of the particle $\hat{x}_{n+1}^{(i)}$ appears with the (unnormalized) probability $\exp\big(-\frac{\xi^T\xi}{2}\big)J_{\psi^{n+1,i}}^{-1}$, where $J_{\psi^{n+1,i}}$ denotes the Jacobian of $\psi^{n+1,i}$, and the weight on $\hat{x}_{n+1}^{(i)}$ equals $\exp\big(-\frac{\xi^T\xi}{2}\big)\exp(-\gamma^{n+1,i})$ [28], i.e.

$$\hat{\alpha}_{n+1}^{(i)} := \exp\big(-\frac{\xi^T\xi}{2}\big)\exp(-\gamma^{n+1,i}). \tag{14}$$

To generate equally weighted particles, we apply the importance sampling method to resample particles $\{\hat{x}_{n+1}^{(i)}\}_{i=1}^{N_p}$. Specifically, we normalize the weights $\{\hat{\alpha}_{n+1}^{(i)}\}_{i=1}^{N_p}$ obtained in (14) so that $\sum_{i=1}^{N_p}\hat{\alpha}_{n+1}^{(i)} = 1$. For each of N_p random numbers ζ_k, $k = 1, 2, \cdots, N_p$, drawn from the uniform distribution on $[0, 1]$, we choose a point $x_{n+1}^{(k)}$ randomly from the particle set $\{\hat{x}_{n+1}^{(i)}\}_{i=1}^{N_p}$ such that

$$\sum_{j=1}^{k-1}\hat{\alpha}_{n+1}^{(j)} < \zeta_k < \sum_{j=1}^{k}\hat{\alpha}_{n+1}^{(j)}.$$

Then the particle set $\{x_{n+1}^{(i)}\}_{i=1}^{N_p}$ follows the importance distribution described by the weighted sample pairs $\{(\hat{x}_{n+1}^{(i)}, \hat{\alpha}_{n+1}^{(i)})\}_{i=1}^{N_p}$, and each particle $x_{n+1}^{(i)}$ has an equal weight. Note that the particle set $\{x_{n+1}^{(i)}\}_{i=1}^{N_p}$ resampled from $\{\hat{x}_{n+1}^{(i)}\}_{i=1}^{N_p}$ describes the conditional distribution of the target better than the particles resampled from $\{\tilde{x}_{n+1}^{(i)}\}_{i=1}^{N_p}$ since $\{\hat{x}_{n+1}^{(i)}\}_{i=1}^{N_p}$ are already in the high density region through the implicit sampling.

Theoretically, the implicit particle filter method can generate equally weighted particles that follow the desired posterior distribution if the (nonlinear) equation (13) can be effectively solved, and the performance of the implicit particle filter is based on the performance of the optimization procedure that solves the equation. However, the complexity of the equation (13) depends on the function F, which relies on both the transition probability and the likelihood function (as indicated in (12)). In this way, the implicit particle filter could be computationally expensive and challenging when the dynamical model is not well-aligned with the observational data.

4 Drift homotopy implicit particle filter

The main effort of this work is to combine advantages of the drift homotopy particle filter and the implicit particle filter to construct a drift homotopy implicit particle filter (DHIPF) method, which can effectively use observational data and efficiently generate particles that follow the filtering density of the target. The general framework of our approach adopts the drift homotopy procedure in the drift homotopy particle filter. Instead of using MCMC as a sampling method in the drift homotopy particle filter, we carry out the "implicit sampling" procedure introduced the implicit particle filter to generate the desired particles efficiently.

To proceed, we recall that in the drift homotopy particle filter, the drift homotopy sequence (10) builds a bridge that connects the intermedia dynamics b to the original state dynamics f. With the observational data incorporated through likelihood (as described in (11)), the drift homotopy distribution $p_l(X^*|x_n^{(i)}, Y_{1:n+1})$ in each drift homotopy step is proportional to $p_l(X^*|x_n^{(i)})p(Y_{n+1}|X^*)$. In the DHIPF, instead of simply using the drift homotopy dynamics as a bridge to transport samples, we consider the random variable X in the drift homotopy dynamics (10) as the state of a nonlinear filtering problem at the time instant n and consider X^* as the state at the time instant $n+1$. Therefore, the drift homotopy distribution $p_l(X^*|x_n^{(i)}, Y_{1:n+1})$ is equivalent to the filtering density of the following nonlinear filtering problem

$$\begin{aligned} \tilde{X}_{n+1} &= (1-\beta_l)b(\tilde{X}_n) + \beta_l f(\tilde{X}_n) + \sigma_n w_n, && \text{(State)} \\ \tilde{Y}_{n+1} &= g(\tilde{X}_{n+1}) + v_n, && \text{(Observation)} \end{aligned} \tag{15}$$

given that the state $\tilde{X}_n$ is chosen as a particle $x_n^{(i)}$ in the previous particle cloud $\{x_n^{(i)}\}_{i=1}^{N_p}$ of the original nonlinear filtering problem (2) and the observation $\tilde{Y}_{n+1}$ is taken as the observational data Y_{n+1}. In other words, we have

$$p_l(X^*|x_n^{(i)}, Y_{1:n+1}) = p_l(\tilde{X}_{n+1}|\tilde{X}_n, \tilde{Y}_{n+1})\big|_{\tilde{X}_n = x_n^{(i)}, \tilde{Y}_{n+1} = Y_{1:n+1}}.$$

Then, the implicit particle filter method can be applied to solve the nonlinear filtering problem (15) and produce a particle that follows the drift homotopy distribution $p_l(X^*|x_n^{(i)}, Y_{1:n+1})$.

Specifically, for an appropriately chosen reference random variable ξ, we solve for the function $\psi_l^{n+1,i}(\xi)$ that connects highly probable values of ξ to highly probable values of $\hat{x}_{n+1,l}^{(i)}$, i.e. $\hat{x}_{n+1,l}^{(i)} = \psi_l^{n+1,i}(\xi)$, where $\hat{x}_{n+1,l}^{(i)}$ is a particle that follows the distribution $p_l(\tilde{X}_{n+1}|\tilde{X}_n, \tilde{Y}_{n+1})\big|_{\tilde{X}_n = x_n^{(i)}, \tilde{Y}_{n+1} = Y_{1:n+1}}$. To this end, we define a function $F_l^{n+1,i}$ (corresponding to the particle $x_n^{(i)}$) by

$$\exp\Big(-F_l^{n+1,i}(\hat{x}_{n+1,l}^{(i)})\Big) := p_l(\hat{x}_{n+1,l}^{(i)}|x_n^{(i)})p(Y_{n+1}|\hat{x}_{n+1,l}^{(i)}), \tag{16}$$

where p_l is the transition probability of the l-th drift homotopy dynamics. Then, we solve the following equation

$$F_l^{n+1,i}\Big(\psi_l^{n+1,i}(\xi)\Big) - \gamma_l^{n+1,i} = \frac{\xi^T\xi}{2} \tag{17}$$

to obtain the function $\psi_l^{n+1,i}$, where $\gamma_l^{n+1,i} := \min F_l^{n+1,i} + \lambda$ is the factor that makes (17) solvable as we introduced in the equation (13). Then, with the connection function $\psi_l^{n+1,i}$ solved through the equation (17), we can generate the position of the particle $\hat{x}_{n+1,l}^{(i)}$ through the expression $\exp\big(-\frac{\xi^T\xi}{2}\big)J^{-1}_{\psi_l^{n+1,i}}$ for a pre-chosen sample that follows ξ.

Similar to the procedure that solves the equation (13) in the implicit particle filter, we use an optimization-based approach to determine $\psi_l^{n+1,i}$ (for a given sample drawn from ξ). Note that the drift homotopy dynamics morph gradually from the intermediate drift b to the original dynamical model f. Therefore, the transition probabilities $\{p_l\}_{l=1}^L$ between two successive drift homotopy levels have similar distributions. Hence the values of implicit functions $\{\psi_l^{n+1,i}\}_{l=1}^L$ should be close for two successive drift homotopy levels. To take the advantage of those bridging drift homotopy dynamics, we use the sample $\hat{x}_{n+1,l-1}^{(i)}$ obtained in the $l-1$-th drift homotopy level as the initial condition for the optimization procedure when solving for $\psi_l^{n+1,i}$. As a result, the optimization for solving $\psi_l^{n+1,i}$ convergences quickly due to the "good" initial condition and the implicit particle filter can be carried out efficiently.

In the last homotopy level L, the drift homotopy dynamics become the original state dynamics f. Therefore, once we obtain the function $\psi_L^{n+1,i}$ through the implicit sampling procedure (16) - (17), we obtain the sample $\hat{x}_{n+1,L}^{(i)}$ that follows the desired filtering density $p(X_{n+1}|x_n^{(i)}, Y_{1:n+1})$ given the particle $_n^{(i)}$ and the fact $p_L(X^*|x_n^{(i)}) = p_f(X_{n+1}|x_n^{(i)})$.

Our DHIPF algorithm is summarized in Table 1. Based on the above discussions for DHIPF, we can see that the optimization based implicit sampling procedure can generate the drift homotopy sample $\hat{x}_l^{(i)}$ much more efficiently – compared with the standard MCMC sampling method. On the other hand, the drift homotopy procedure creates a bridge that connects an intermediate dynamical function and the original state

Table 1:

Algorithm: *Drift homotopy implicit particle filter (DHIPF)*

Initialize the particle cloud $\{x_0^{(i)}\}_{i=1}^{N_p}$, the number of drift homotopy levels L with the intermediate drift function b and the constant sequence $\{\beta_l\}_{l=0}^{L}$, and the reference random variable ξ for the implicit particle filter procedure.

while $n = 0, 1, 2, \cdots$, **do**

for particles $i = 1, 2, \cdots, N_p$,

for drift homotopy levels $l = 0, 1, 2, \cdots, L-1$,

- Construct the drift homotopy dynamics (10);
- Solve for $\psi_l^{n+1,i}$ in the equation (17) with the initial guess $\hat{x}_{n+1,l}^{(i)}$;
- Generate the sample $\hat{x}_{n+1,l+1}^{(i)}$ through $\exp\big(-\frac{\xi^T\xi}{2}\big)J^{-1}_{\psi_l^{n+1,i}}$;

end for

end for

The particles $\{x_{n+1}^{(i)}\}_{i=1}^{N_p} := \{\hat{x}_{n+1,L}^{(i)}\}_{i=1}^{N_p}$ provide an empirical distribution for the filtering density $p(X_{n+1}|Y_{1:n+1})$

end while

dynamics. Since two successive drift homotopy dynamics are similar, solutions of drift homotopy filtering problems (15) change gradually to the filtering density of the original nonlinear filtering problem. Therefore, it is easy to achieve the optimality condition in the implicit particle filter , and hence the implicit particle filter can be implemented efficiently under our DHIPF framework. Moreover, the intermediate dynamics b in the DHIPF is designed in a way so that the likelihood function would dominate the first few drift homotopy steps. Then, the drift homotopy procedure incorporates the original filtering dynamics and let the dynamical model combine with the observational data. In other words, the drift homotopy procedure aims to construct the desired filtering density starting from the likelihood instead of starting from the predicated model, which is typically implemented by most optimal filtering methods. In this way, the DHIPF could endow the implicit particle filter the mechanism that trusts the observational data first.

5 Numerical experiments

In this section, we present two benchmark numerical examples to demonstrate the performance of our DHIPF method. In the first example, we track the state of a stochastic dynamical system driven by the double well potential. To demonstrate the advanta-

geous performance of DHIPF, we compare our method with the implicit particle filter and the standard drift homotopy particle filter (with MCMC sampling) – along with other "state-of-the-art" methods. In the second example, we solve a Lorenz attractor problem. The target state that we estimate is driven by the Lorenz 63 model, which is a well-known chaotic dynamical system. We show that our method can capture the unpredicted chaotic behavior of the model by effective processing of observational data. All the numerical experiments are carried out on an Intel Core i7-670HQ 2.6GHz CPU.

5.1 Double well potential

The double well potential is an important quartic model in quantum mechanics and quantum field theory, and models derived from the double well potential have been widely used in nano-phase materials [18]. The potential U, described by

$$U(x) = \frac{\alpha}{4}(x^4 - 2x^2),$$

has two stable positions at $x = 1$ and $x = -1$, where α is the model parameter that determines the "depth" of potential wells. While a particle is at positions other than 1 and -1, it will be pushed by a force with the magnitude of $U'(x)$ towards one of the stable positions. In this example, we aim to estimate the state of a stochastic dynamical system driven by the double well potential, i.e.

$$dX_t = -\alpha(X_t^3 - X_t)dt + \sigma dW_t, \qquad 0 \le t \le T$$

and the data that we use to track the target state are direct observations on X, which are perturbed by Gaussian noises with standard deviation R.

We consider the following discretized nonlinear filtering problem

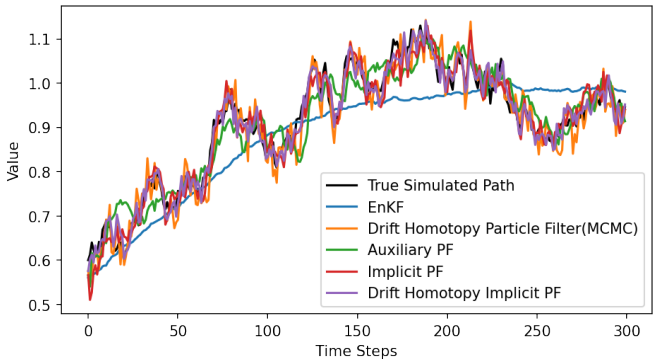
$$\begin{aligned} X_{n+1} =& X_n - \alpha(X_n^3 - X_n)\Delta t + \sigma w_n, \\ Y_{n+1} =& X_{n+1} + R v_n, \end{aligned} \tag{18}$$

where w_n and v_n are two independent Gaussian random variables, and we track the state of X for 300 time steps with stepsize $\Delta t = 0.01$. In this example, we compare our DHIPF with four most successful nonlinear filtering methods: the auxiliary particle filter (APF), the ensemble Kalman filter (EnKF), the implicit particle filter and the drift homotopy particle filter (DHPF), where IPF and DHPF (implemented by MCMC sampling) are fundamental components that we use to construct our DHIPF. For all the particle filters, we use 20 particles to describe the one-dimensional state distribution and we use an ensemble of 200 Kalman filter samples in the EnKF. Also, we choose the total number of drift homotopy steps to be $L = 2$, i.e. we use three intermediate dynamical systems to transport particles.

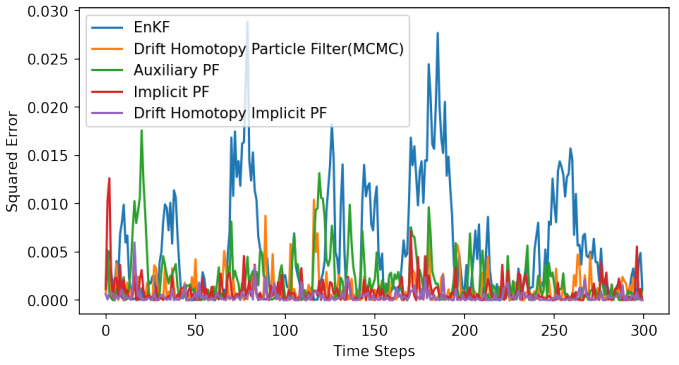
To provide a comprehensive demonstration of the performance of all the nonlinear filtering methods, we solve the double well potential tracking problem for three different cases.

Case 1.

In this case, we choose the parameters for the double well potential tracking problem as $\alpha = 1$, $\sigma = 1.5$, and $R = 1.5$, and the initial state X is set to be $X_0 = 0.6$. In Figure 1



(a) Tracking performance.



(b) Estimation errors.

Figure 1: Double well potential case 1: $\alpha = 1$, $\sigma = 1.5$, $R = 1.5$

(a), we present the state estimation for the target state obtained by different nonlinear filtering methods, where the black curve is the true simulated state and the colored curves are estimates. From this subplot, we can see that all the methods captured the main trend of the state while the EnKF failed to capture detailed behaviors of the target. To show more details of estimation accuracy, in Figure 1 (b) we plot errors of each method in estimating the state X. From this subplot, we can see that the IPF, the DHPF and the DHIPF can give very accurate estimates for the state, the APF occasionally suffers large errors, and the EnKF has the worst performance.

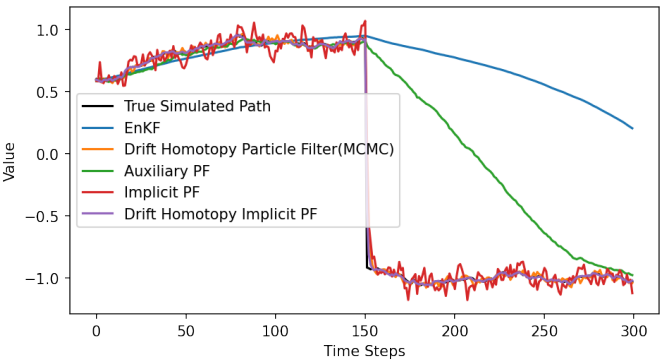
Table 2: Example 1. Performance comparison for Case 1

	APF	EnKF	IPF	DHPF	**DHIPF**
CPU Time	9.703	0.365625	0.0938	48.563	**0.312**
MSE	$2.03E-3$	$5.22E-3$	$1.10E-3$	$5.92E-4$	$\mathbf{4.60E-4}$

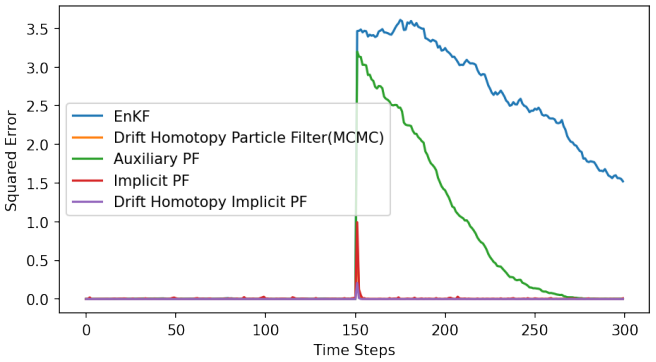
In order to further demonstrate the performance of all the methods, we present the accumulated mean square error (MSE) of each method along with its CPU time in solving this double well potential tracking problem in Table 2. From the comparison table, we can see that the DHPF (with MCMC sampling) and the DHIPF have the lowest estimation errors. However, the DHIPF spends much less CPU time compared with DHPF due to the efficient implicit sampling procedure.

Case 2.

In this case, we let $\alpha = 1$, $\sigma = 1$, and $R = 1$, and the initial state X is set to be $X_0 = 0.6$. Different from the first case, we observe that the real state switched from the stable position $X = 1$ to the stable position $X = -1$ after approximately 150 tracking steps. In physics, this kind of switch may be caused by some unexpected external force or some extreme diffusion activities. We use this experiment to demonstrate the "data first" advantage of drift homotopy procedure, and we present the tracking performance and the estimation error of each method in Figure 2 (a) and (b), respectively. From this



(a) Tracking performance.



(b) Estimation errors.

Figure 2: Double well potential case 2: $\alpha = 1$, $\sigma = 1$, $R = 1$ with state switch

figure, we can see that the EnKF and APF could not follow effectively the rapid change in the target state. On the other hand, the IPF, the DHPF and the DHIPF captured the switch of the real state effectively while the IPF has slightly higher error at the time of switch. In Table 3, we present the CPU time and the MSE of each method. We

Table 3: Example 1. Performance comparison for Case 2

	APF	EnKF	IPF	DHPF	**DHIPF**
CPU Time	9.391	0.578	0.109	43.344	**0.297**
MSE	$6.29E-1$	1.36	$5.28E-3$	$1.78E-3$	$\mathbf{1.08E-3}$

can see from this table that due to the unexpected switch of the target state, both the APF and the EnKF have high estimation errors. The DHPF and the DHIPF have very low MSEs in tracking the state, and the DHIPF has much lower computational cost compared with DHPF.

Case 3.

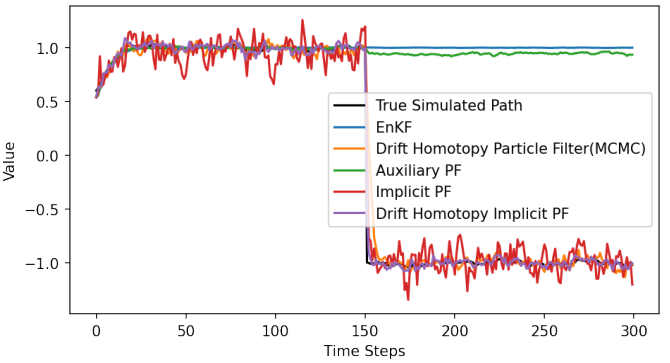
In this case, we also consider state switch during the tracking period. This time, we choose parameters $\alpha = 10$, $\sigma = 1$, and $R = 2$ for the nonlinear filtering problem, where the large parameter α indicates that the potential wells are very "deep". In Figure 3, we present the tracking performance and the estimation error of each method. We can see from this figure that both the EnKF and the APF completely failed to capture the state switch. On the other hand, IPF, DHPF and DHIPF captured the state switch well. In

Table 4: Example 1. Performance comparison for Case 3

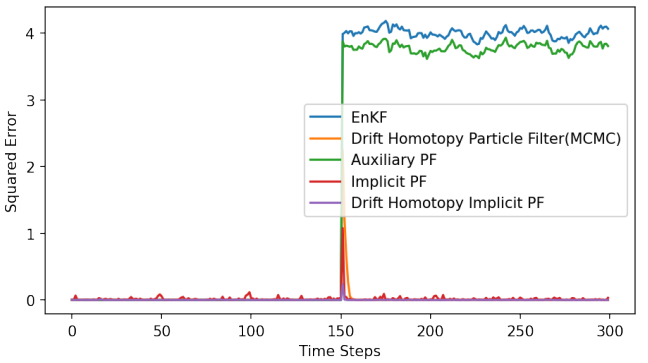
	APF	EnKF	IPF	DHPF	**DHIPF**
CPU Time	9.563	0.453	0.156	50.188	**0.422**
MSE	1.80	1.99	$5.02E-3$	$1.35E-2$	$\mathbf{1.59E-3}$

Table 4, we present the CPU time and the MSE of each method. We can see that DHIPF has the lowest accumulative error, and its computational cost is comparable to IPF.

From the above numerical experiments, we can see that our DHIPF method outperforms EnKF in accuracy, and it steadily outperforms APF and DHPF in both efficiency and accuracy. In comparison with IPF, DHIPF typically has higher accuracy. On the other hand, the computational cost for DHIPF is slightly higher than IPF since DHIPF requires several implicit sampling procedures. To give more details of advantageous performance of DHIPF compared with IPF, in the next numerical example we focus on the comparison between DHIPF and IPF.



(a) Tracking performance.



(b) Estimation errors.

Figure 3: Double well potential case 3: $\alpha = 10$, $\sigma = 1$, $R = 2$ with state switch

5.2 Lorenz attractor

In this example, we solve a Lorenz attractor problem, which has wide applications in weather forecasting and climate prediction. The Lorenz dynamics that we consider is the Lorenz 63 model, which is described by

$$Lz(\mathbf{x}) = \begin{pmatrix} a_1(y-x) \\ a_2 x - y - xz \\ xy - a_3 z \end{pmatrix}, \tag{19}$$

where $\mathbf{x} = (x, y, z)$ is a three-dimensional vector, a_1 is the Prandtl number, a_2 is a normalized Rayleigh number and a_3 is a non-dimensional wavenumber. The nonlinear filtering problem corresponding to the Lorenz 63 model (19) is given by

$$\begin{aligned} dX_t &= dLz(t)dt + \sigma dW_t, \\ dY_t &= X_t dt + R dV_t, \end{aligned} \tag{20}$$

where W and V are two standard Brownian motions, σ is the diffusion coefficient, which decides the size of noises that perturb the state process, and R is the coefficient for observational noises. In our numerical experiments, we choose $a_1 = 10$, $a_2 = 28$ and $a_3 = 8/3$, which will result chaotic behavior of the state process, and we let $\sigma = \mathbf{I}_{3\times 3}$, $R = \mathbf{I}_{3\times 3}$.

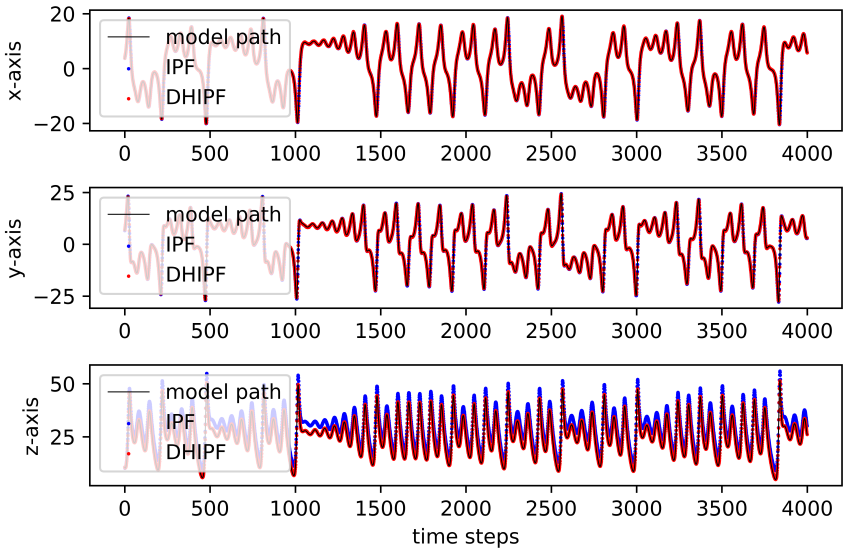


Figure 4: Tracking performance for 4000 steps.

In the first numerical experiment, we estimate the state X over the time interval $[0, 40]$ with step-size 0.01, i.e. 4000 steps, and we use 10 particles to implement both DHIPF and IPF. In Figure 4, we present the tracking performance of DHIPF and IPF with respect to each dimension. The black curve is the true target state, the blue dotted curve is the estimate obtained by IPF, and the red dotted curve is the estimate obtained by DHIPF. From this figure, we can see that generally DHIPF and IPF provide good estimates for the target state. Specifically, they both give accurate estimates in x and y directions, and DHIPF is consistently more accurate than IPF in the z direction. To take a closer look at the accuracy between DHIPF and IPF, we plot squared errors combining all three directions in Figure 5. From this figure, wes see that DHIPF is more accurate than IPF over the entire tracking period.

In our second numerical experiment in this Lorenz attractor example, we present the tracking performance of DHIPF and IPF when the true state of the Lorenz dynamics moves rapidly between time steps 300 and 400. Such rapid motion are typically caused by the chaotic nature of the Lorenz model, which is often observed when predicting weather in real time. To implement DHIPF and IPF, we use 30 particles to adjust the possible fast state change and we estimate the target state for 600 time steps.

In Figure 6, we show the estimation performance of DHIPF and IPF in each dimension. We can see from the figure that DHIPF accurately captured the true state of the

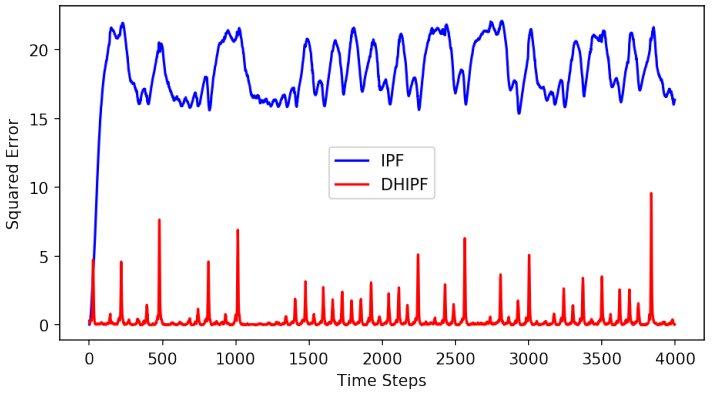


Figure 5: Tracking errors for 4000 steps.

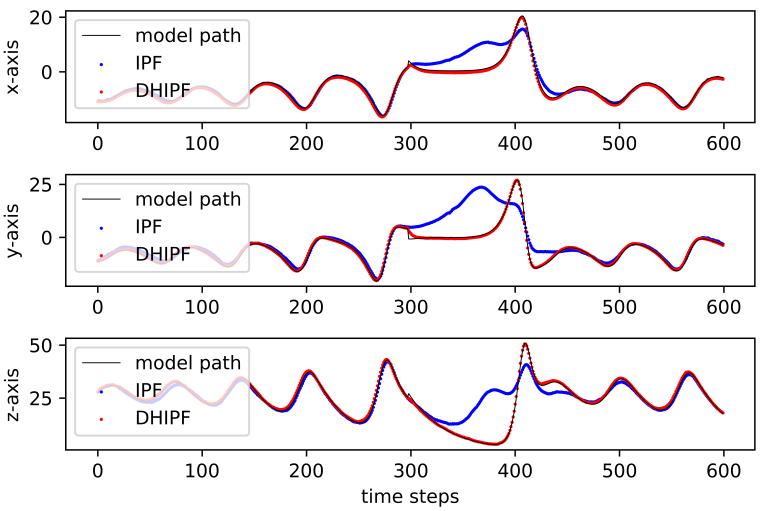


Figure 6: Tracking performance with rapid change in the state.

Lorenz dynamics even during the rapid motion period. On the other hand, IPF could only follow the trend of the target motions and it took IPF over 100 steps to recover good estimates. In Figure 7, we plot the squared errors combining three dimensions over the tracking time, and we can see clearly that IPF suffers large errors from time step 300 to approximately time step 450 while DHIPF only has a small spike in estimation errors to adjust the chaotic behavior of the model.

To further demonstrate the advantageous performance of DHIPF over IPF, we assume that there are gaps between model simulations and observations. This could reflect the situation that the data are collected occasionally, and such a situation occurs frequently in practice. In this experiment, we use 50 particles for both DHIPF and

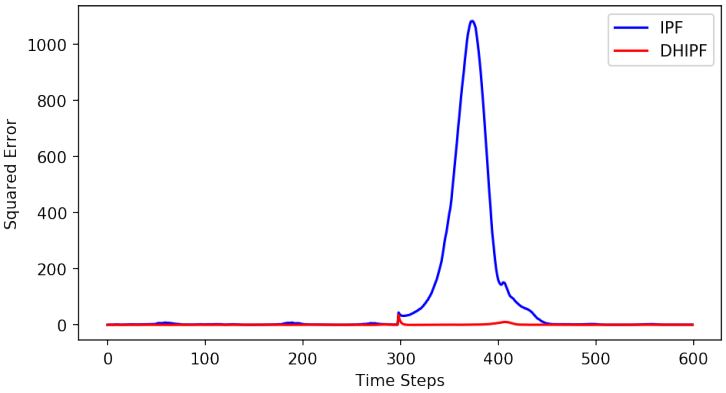


Figure 7: Tracking errors with rapid change in the state.

IPF, and we track the target state with 1000 simulation steps. In Figure 8, we solve

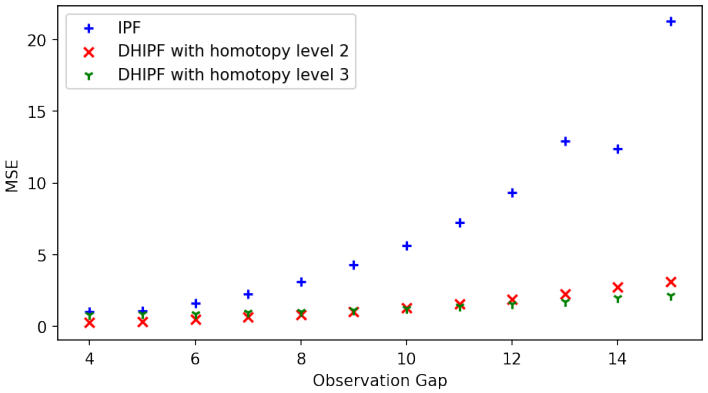


Figure 8: Mean square errors with respect to observation gaps.

the Lorenz attractor problem repeatedly over 20 times and plot the MSEs among all 20 repeated tests of each method with respect to observation gaps, where the blue markers are MSEs of IPF, the red markers are MSEs of DHIPF with 2 drift homotopy levels, i.e. $L = 2$, and the green markers are MSEs of DHIPF with 3 drift homotopy levels, i.e. $L = 3$. From this figure, we see that the errors of IPF increase as observations become sparser. On the other hand, although DHIPF has larger errors when observations are sparser, it is not as sensitive as IPF with respect to observation gaps, and DHIPF always has accurate estimates for the state.The reason why DHIPF has better performance in this "observation gap experiment" is that the drift homotopy procedure allows us to process the observational data first before we incorporate the dynamical model. As a result, in the case that data are hard to collect, which means each set of observational data is "more valuable", DHIPF can utilize the observational data more effectively and therefore obtain better results. Moreover, we see from this figure that

for smaller observation gaps, DHIPF with 2 homotopy levels has similar performance to 3 homotopy levels. When the observation gap is getting larger, more homotopy levels bring more accurate results. This also supports the utility of the homotopy procedure.

6 Acknowledgement

This work is partially supported by the Scientific Discovery through Advanced Computing (SciDAC) program funded by U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research through FASTMath Institute and CompFUSE project. The second author also acknowledges support by U.S. National Science Foundation under Contract DMS-1720222.

References

[1] C. Andrieu, A. Doucet, and R. Holenstein. Particle markov chain monte carlo methods. *J. R. Statist. Soc. B*, 72(3):269–342, 2010.

[2] C. Andrieu and G. O. Roberts. The pseudo-marginal approach for efficient monte carlo computations. *Ann. Statist.*, 37(2):697–725, 2009.

[3] R. Archibald, F. Bao, and X. Tu. A direct filter method for parameter estimation. *J. Comput. Phys.*, 398:108871, 17, 2019.

[4] F. Bao, Y. Cao, and X. Han. An implicit algorithm of solving nonlinear filtering problems. *Communications in Computational Physics*, 16:382–402, 2014.

[5] F. Bao, Y. Cao, and X. Han. Forward backward doubly stochastic differential equations and optimal filtering of diffusion processes. *Communications in Mathematical Sciences*, 18(3):635–661, 2020.

[6] F. Bao, Y. Cao, and W. Zhao. Numerical solutions for forward backward doubly stochastic differential equations and zakai equations. *International Journal for Uncertainty Quantification*, 1(4):351–367, 2011.

[7] F. Bao, Y. Cao, and W. Zhao. A first order semi-discrete algorithm for backward doubly stochastic differential equations. *Discrete and Continuous Dynamical Systems-Series B*, 5(2):1297 – 1313, 2015.

[8] Feng Bao, Richard Archibald, and Petro Maksymovych. Backward SDE filter for jump diffusion processes and its applications in material sciences. *Communications in Computational Physics*, 27:589–618, 2020.

[9] Feng Bao, Yanzhao Cao, and Hongmei Chi. Adjoint forward backward stochastic differential equations driven by jump diffusion processes and its application to nonlinear filtering problems. *Int. J. Uncertain. Quantif.*, 9(2):143–159, 2019.

[10] Feng Bao, Yanzhao Cao, Amnon Meir, and Weidong Zhao. A first order scheme for backward doubly stochastic differential equations. *SIAM/ASA J. Uncertain. Quantif.*, 4(1):413–445, 2016.

[11] Feng Bao, Yanzhao Cao, Clayton Webster, and Guannan Zhang. A hybrid sparse-grid approach for nonlinear filtering problems based on adaptive-domain of the Zakai equation approximations. *SIAM/ASA J. Uncertain. Quantif.*, 2(1):784–804, 2014.

[12] Feng Bao, Yanzhao Cao, and Weidong Zhao. A backward doubly stochastic differential equation approach for nonlinear filtering problems. *Commun. Comput. Phys.*, 23(5):1573–1601, 2018.

[13] Feng Bao and Vasileios Maroulas. Adaptive meshfree backward SDE filter. *SIAM J. Sci. Comput.*, 39(6):A2664–A2683, 2017.

[14] A. J. Chorin and X. Tu. Implicit sampling for particle filters. *Proc. Nat. Acad. Sc. USA*, 106:17249–17254, 2009.

[15] D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *IEEE Trans. Sig. Proc.*, 50(3):736–746, 2002.

[16] Dan Crisan. Exact rates of convergence for a branching particle approximation to the solution of the Zakai equation. *Ann. Probab.*, 31(2):693–718, 2003.

[17] A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: fifteen years later. *The Oxford handbook of nonlinear filtering*, 2011.

[18] O. Dyck, M. Ziatdinov, S. Jesse, F. Bao, A. Yousefzadi Nobakht, A. Maksov, B.G. Sumpter, R. Archibald, K.J.H. Law, and S.V. Kalinin. Probing potential energy landscapes via electron-beam-induced single atom dynamics. *Acta Materialia*, 203:116508, 2021.

[19] G. Evensen. *Data assimilation: the ensemble Kalman filter*. Springer, 2006.

[20] G. Evensen. The ensemble Kalman filter for combined state and parameter estimation: Monte Carlo techniques for data assimilation in large systems. *IEEE Control Syst. Mag.*, 29(3):83–104, 2009.

[21] Emmanuel Gobet, Gilles Pagès, Huyên Pham, and Jacques Printems. Discretization and simulation of the Zakai equation. *SIAM J. Numer. Anal.*, 44(6):2505–2538 (electronic), 2006.

[22] N.J Gordon, D.J Salmond, and A.F.M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE PROCEEDING-F*, 140(2):107–113, 1993.

[23] S.J. Julier and J.K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92:401–422, 2004.

[24] Kai Kang, Vasileios Maroulas, Ioannis Schizas, and Feng Bao. Improved distributed particle filters for tracking in a wireless sensor network. *Comput. Statist. Data Anal.*, 117:90–108, 2018.

[25] H. R. Kunsch. Particle filters. *Bernoulli*, 19(4):1391–1403, 2013.

[26] François Le Gland. Time discretization of nonlinear filtering equations. In *Proceedings of the 28th IEEE Conference on Decision and Control, Vol. 1–3 (Tampa, FL, 1989)*, pages 2601–2606, New York, 1989. IEEE.

[27] Vasileios Maroulas and Panos Stinis. Improved particle filters for multi-target tracking. *Journal of Computational Physics*, 231(2):602 – 611, 2012.

[28] M. Morzfeld, X. Tu, E. Atkins, and A. J. Chorin. A random map implementation of implicit filters. *J. Comput. Phys.*, 231(4):2049–2066, 2012.

[29] Michael K. Pitt and Neil Shephard. Filtering via simulation: auxiliary particle filters. *J. Amer. Statist. Assoc.*, 94(446):590–599, 1999.

[30] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson. Obstacles to high-dimensional particle filtering. *Mon. Wea. Rev.*, 136:4629–4640, 2008.

[31] Taek Song and J. Speyer. A stochastic analysis of a modified gain extended kalman filter with applications to estimation with bearings only measurements. *IEEE Transactions on Automatic Control*, 30(10):940–949, 1985.

[32] Xin T. Tong, Andrew J. Majda, and David Kelly. Nonlinear stability and ergodicity of ensemble based Kalman filters. *Nonlinearity*, 29(2):657–691, 2016.

[33] P. J. van Leeuwen. Nonlinear data assimilation in geosciences: an extremely efficient particle filter. *Q. J. Roy. Meteor. Soc.*, 136(653):1991–1999, 2010.

[34] Bin Wang, Xiaolei Zou, and Jiang Zhu. Data assimilation and its applications. *Proceedings of the National Academy of Sciences*, 97(21):11143–11144, 2000.

[35] Moshe Zakai. On the optimal filtering of diffusion processes. *Z. Wahrscheinlichkeitstheorie und Verw. Gebiete*, 11:230–243, 1969.