18th International Conference on Information Fusion
Washington, DC - July 6-9, 2015

A Learning Drift Homotopy Particle Filter

Vasileios Maroulas
Dept of Math
University of Tennessee
Knoxville, Tennessee 37996
Email: vmaroula@utk.edu

Kai Kang
Dept of Math
University of Tennessee
Knoxville, Tennessee 37996
Email:kang@math.utk.edu

Ioannis D. Schizas
Dept of EE
University of Texas
Arlington, Texas 76010
Email: schizas@uta.edu

Michael W. Berry
Dept of EECS
University of Tennessee
Knoxville, Tennessee 37996
Email: mberry@eecs.utk.edu

***Abstract*—In this paper, we design a learning drift homotopy particle filter algorithm. We employ the drift homotopy technique in the extra Markov Chain Monte Carlo move after the resampling step of the generic particle filter algorithm to efficiently resolve the degeneracy of the algorithm. In this work, we use the effective sample size as a learning parameter to control the levels of drift homotopy which need to be considered in each time step. The proposed algorithm adjusts the number of levels of drift homotopy and reduces its computational time without undermining the accuracy of estimation. We test the algorithm on two synthetic problems, a partially observed diffusion in a double well potential and a multi-target tracking setting.**

***Index Terms*—Learning methods, DDDAS, drift homotopy, Markov Chain Monte Carlo, particle filtering, sequential Bayesian estimation.**

I. INTRODUCTION

Many spatiotemporal phenomena require the estimation of a partially observed model, which captures the dynamics of the phenomenon under investigation. These dynamics are typically observed as a result of collected data. For example, objects are continuously moving in a field and a distributed surveillance sensor network collects data about the position of moving threats [4], [20], [24], [27].

Precisely, a system of interest can be modeled as a state space hidden Markov model (HMM), which is engaged with a pertinent likelihood function. Employing a Bayesian framework, inference on the unknown quantities are deduced by providing a posterior distribution at any given time, called a filtering distribution. On the other hand, the theoretical solution is often times intractable and thus several methods have been investigated for approximating the filtering distribution.

Particle filtering is a popular method which approximates the filtering posterior distribution by a set of weighted samples generated by a proper distribution which facilitates the sampling called an importance distribution. Its choice depends on the problem under investigation and several authors have suggested a variety of methods, e.g. see the partial list [5], [12], [11], [25], [26]. However, most of the particles have a negligible weight which implies that their contribution to the approximation is insignificant. A resampling step is in turn used to mitigate the increasing variance of the weights as time evolves. On the other hand, even with this extra step the degeneracy of particle filtering still exists.

In this paper, we bypass this problem by proposing a learning drift homotopy particle filter algorithm. We engage the idea of appending an extra Markov Chain Monte Carlo (MCMC) step after the resampling step, see e.g. [29], which aims to move the particles to statistically significant regions. In contrast, the issue with the extra MCMC step is to preserve the nature of the posterior distribution and the speed of the algorithm's convergence. Therefore, we use a learning drift homotopy technique to achieve these goals.

Drift homotopy strategies consider a sequence of stochastic dynamics with drifts which interpolate between the original and modified drifts. The interpolation engages several levels, $\ell = 0, \dots, L$, for which at $\ell = 0$ the modified dynamics are in effect, as opposed at $\ell = L$ where the original ones are. All intermediate levels, $0 < \ell < L$ are auxiliary and basically facilitate the MCMC sampling. In other words, one constructs paths for each stochastic equation in the sequence at level ℓ by using an appropriate MCMC scheme with initial condition the stochastic equation at level $\ell - 1$. This allows to gradually morph a path with a low weight to a path with a significant weight while respecting the nature of the filtering distribution [17], [18]. At the same time, a learning parameter in the drift homotopy technique is used to reduce the computational cost. It plays the role of controlling the levels of drift homotopy required in the approximation of the filtering distribution. Adopting the DDDAS framework [9], at each time step, the learning parameter is configured automatically by measuring the degree of degeneracy of the particle filter. This mechanism is able to decrease the computational cost without compromising the accuracy of estimation. We test this novel method in the problem of a partially observed diffusion in a double well potential, and in a multi-target tracking scenario.

The paper is organized as follows. Section II describes the particle filter with MCMC moves. Section III provides a detailed illustration of integrating the learning drift homotopy technique in the appended MCMC step. Section IV exposes our numerical results on two examples, and at last a conclusion is offered at Section V.

II. PARTICLE FILTER WITH MARKOV CHAIN MONTE CARLO MOVES

Consider $\{\mathbf{x}_k\}$ the state of a hidden Markov model observed by $\{\mathbf{y}_k\}$ at a given time instant $k \in \mathbb{N} \cup \{0\}$. We assume that at the initial time $k = 0$, $\mathbf{x}_0$ is distributed according to $p(\mathbf{x}_0)$. In addition, the transition density, $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, and the likelihood function, $p(\mathbf{y}_k|\mathbf{x}_k)$, are given. Our goal is to compute the posterior distribution $p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$ or a pertinent moment, $\mathbb{E}[f(\mathbf{x}_{0:k})]$, where $\mathbf{x}_{0:k} = \{\mathbf{x}_0, \cdots, \mathbf{x}_k\}$ denotes the discrete history of the process up to a given time k and $\mathbb{E}$ denotes expectation. Employing a recursive framework with Bayesian considerations, e.g. see [2], yields the basis of sequential Monte Carlo (SMC) methods given by eq. (1)

$$p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) = p(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1})\frac{p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{y}_k|\mathbf{x}_k)}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} \quad (1)$$

Under general conditions, a closed form of the posterior distribution is not tractable analytically based on eq. (1). Particle filter, which is a sequential importance sampling technique, approximates the filtering distribution by a set of weighted particles. Imagine that the expectation, $\mathbb{E}\mathbf{x}_{0:k}$ with respect to the filtering distribution $p = p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$, needs to be estimated. Taking into account an importance density, $q(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$, which facilitates the sampling, one may compute the expectation as follows,

$$\mathbb{E}_p(\mathbf{x}_{0:k}) = \frac{\int \mathbf{x}_{0:k}\frac{p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})}{q(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})}q(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})d\mathbf{x}_{0:k}}{\int \frac{p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})}{q(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})}q(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})d\mathbf{x}_{0:k}} \quad (2)$$

Considering i.i.d. samples, $\mathbf{x}^i_{0:k}, i = 1, \ldots, N$, from the importance distribution, $q(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$, the approximating eq. (2) via Monte Carlo integration yields that

$$\mathbb{E}_p(\mathbf{x}_{0:k}) \approx \frac{\sum_{i=1}^N \mathbf{x}^i_{0:k} w^i_k}{\sum_{j=1}^N w^j_k} = \sum_{i=1}^N \mathbf{x}^i_{0:k}\tilde{w}^i_k, \quad (3)$$

where the normalized weights $\tilde{w}^i_k \propto \frac{p(\mathbf{x}^i_{0:k}|\mathbf{y}_{1:k})}{q(\mathbf{x}^i_{0:k}|\mathbf{y}_{1:k})}$ satisfies $\sum_{i=1}^N \tilde{w}^i_k = 1$. Similarly, the filtering distribution can be estimated by $\hat{p}(\mathbf{x}_{0:k}|\mathbf{y}_{1:k}) = \sum_{i=1}^N \tilde{w}^i_k\delta(\mathbf{x}_{0:k} - \mathbf{x}^i_{0:k})$, where $\delta(\cdot)$ is the delta function.

The choice of importance density is not unique. The study in [11] provides the optimal choice for the importance density. Optimality holds in the sense that it minimizes the variance of the weights. The optimal choice does not have an analytic solution in general. However, it can be used as a guide for suboptimal choices. For instance, [13] establishes a Laplace approximation of the optimal approximation and recently [14] generalized this result using a skew-normal approximation. A popular choice is to adopt the transition density, $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, as the importance density. Then the importance weights [2] are propagated recursively according to

$$\tilde{w}^i_k \propto \tilde{w}^i_{k-1}\frac{p(\mathbf{y}_k|\mathbf{x}^i_k)p(\mathbf{x}^i_k|\mathbf{x}^i_{k-1})}{p(\mathbf{x}^i_k|\mathbf{x}^i_{k-1})} = \tilde{w}^i_{k-1}p(\mathbf{y}_k|\mathbf{x}^i_k). \quad (4)$$

After updating the importance weights at any time step k based on (4), resampling step is performed to decrease the impact of the degeneracy of particle filter by removing particles with low weights and replicating the ones with high weights [2], [13], [12], [16], [21]. There are several ways of executing a resampling step in particle filtering. For example, [8] discusses in detail the popular multinomial resampling scheme and suggests alternative resampling frameworks which keep the number of particles constant such as the residual, the stratified and the systematic resampling methods.

However, even with the resampling method, particle filter requires a large number of particles in order to approximate the filtering distribution. The reason is due to the fact that many particles remain in statistically insignificant regions and thus they do not contribute to the approximation of the filtering distribution. Therefore, many authors [15], [29] have suggested an extra MCMC step after the traditional resampling method. For instance, [29] suggested a resampling step which produces copies not only of the good samples according to the current observations, but also of the values (initial conditions) of the samples at the previous observation. These values propagated good samples for the current observations. Although there are several ways to incorporate such a step, it is crucial that the filtering distribution $p(\mathbf{x}_{0:k}|\mathbf{y}_{1:k})$ is preserved. In this paper, we address this point by introducing a learning drift homotopy method as explained in Section III. Algorithm 1 provides the pseudocode of a particle filter enhanced with MCMC moves.

Initialization at time $k = 0$;
for $k = 1 : T$ **do**
  Sample N unweighted samples $\mathbf{x}^i_{0:k-1}$ from $p(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1})$;
  Prediction: Generate N samples, $\tilde{\mathbf{x}}^i_k$ from $p(\mathbf{x}_k|\mathbf{x}^i_{k-1})$ and set $\tilde{\mathbf{x}}^i_{0:k} = (\mathbf{x}^i_{0:k-1}, \tilde{\mathbf{x}}^i_k)$;
  Update: Compute the weights by (4);
  Resampling: Generate N independent uniform random variables $\{\theta^i\}^N_{i=1}$ in $(0, 1)$. For $i, j = 1, \ldots, N$, let $\mathbf{x}^i_{0:k} = \tilde{\mathbf{x}}^j_{0:k}$ where

$$\sum_{l=1}^{j-1} w^l_k \leq \theta^j < \sum_{l=1}^{j} w^l_k$$

  MCMC step: Sample through MCMC the stationary distribution

$$p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{y}_k|\mathbf{x}_k)$$

end

Algorithm 1: Particle filter with MCMC moves

III. Learning Drift Homotopy Particle Filter

Let's consider that the system of interest evolves according to the following discrete dynamics

$$\mathbf{x}_{t+\Delta} = \mathbf{x}_t + a(\mathbf{x}_t)\Delta + \sigma\sqrt{\Delta}\xi_t, \quad (5)$$

where Δ is the sampling time. The drift $a(\cdot)$ is some pertinent function and σ is the diffusion coefficient which depends on the phenomenon, and ξ_t is a Gaussian random variable. The MCMC step of Algorithm 1 involves sampling from the transition density $p(\mathbf{x}_k|\mathbf{x}_{k-1})$. Many phenomena require that the sampling time, Δ, is very small, e.g. in continuous tracking of a fast moving object. On the other hand, the observations cannot be collected at every time step Δ. This yields that the transition density between time $k-1$ and k is written by the following multidimensional integral,

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) = \int \prod_{\lambda=0}^{I-1} p(\mathbf{x}^{\lambda+1}_{k-1}|\mathbf{x}^{\lambda}_{k-1}) \prod_{\lambda=1}^{I-1} d\mathbf{x}^{\lambda}_{k-1}, \quad (6)$$

where $\mathbf{x}^{\lambda}_{k-1} = \mathbf{x}_{k-1} + \lambda\Delta$ is the state of the process for $\lambda = 0, \cdots, I-1$ with the convention that $\mathbf{x}^0_{k-1} = \mathbf{x}_{k-1}$ and $\mathbf{x}^I_{k-1} = \mathbf{x}_k$. In order to avoid the high dimensional integrals of eq. (6), one samples a conditional path from the transition density $p(\mathbf{x}_k, \mathbf{x}_{k-1}^{I-1}, \cdots, \mathbf{x}_{k-1}^{1}|\mathbf{x}_{k-1})$ given below

$$p(\mathbf{x}_k, \mathbf{x}_{k-1}^{I-1}, \cdots, \mathbf{x}_{k-1}^{1}|\mathbf{x}_{k-1}) = \prod_{\lambda=0}^{I-1} p(\mathbf{x}_{k-1}^{\lambda+1}|\mathbf{x}_{k-1}^{\lambda}). \quad (7)$$

We observe that eq. (7) depends on the transition densities $p(\mathbf{x}_{k-1}^{\lambda+1}|\mathbf{x}_{k-1}^{\lambda})$ for $\lambda = 0, \cdots, I-1$. For some dynamical systems, the diffusion coefficient, σ, of eq. (5) is small. This yields a rare transition which in turn may propagate an erroneous estimation of the filtering distribution of eq. (1). We bypass this problem by introducing a novel learning drift homotopy sampling method. Based on the drift homotopy method [28], one considers modified dynamics for the process and engages them in a sequential way with the original dynamics. In other words, the following dynamics are considered

$$\mathbf{x}_{t+\Delta} = \mathbf{x}_t + (1-\epsilon_\ell)b(\mathbf{x}_t)\Delta + \epsilon_\ell a(\mathbf{x}_t)\Delta + \sigma\sqrt{\Delta}\xi_t \quad (8)$$

where $\epsilon_\ell = \frac{\ell}{L}, \ell = 0, \cdots, L$. One may observe that the original dynamics given in eq. (5) are taken into account at terminal level $\ell = L$ and the modified ones alone are considered when $\ell = 0$. The modified dynamics aid the transition from state $\mathbf{x}_{k-1}$ to $\mathbf{x}_k$ which is of paramount importance when a small diffusion coefficient, σ, yields to a rare transition.

The consideration of eq. (8) deduces that instead of sampling directly from eq. (7), one may sample from a set of $L+1$ distributions given as follows

$$\begin{aligned} & p^\ell(\mathbf{x}_k, \mathbf{x}_{k-1}^{I-1}, \cdots, \mathbf{x}_{k-1}^{1}|\mathbf{x}_{k-1})p(\mathbf{y}_k|\mathbf{x}_k) \\ = \; & \prod_{\lambda=0}^{I-1} p^\ell(\mathbf{x}_{k-1}^{\lambda+1}|\mathbf{x}_{k-1}^{\lambda})p(\mathbf{y}_k|\mathbf{x}_k) \end{aligned} \quad (9)$$

where $p^\ell(\mathbf{x}_k, \mathbf{x}_{k-1}^{I-1}, \cdots, \mathbf{x}_{k-1}^{1}|\mathbf{x}_{k-1}), \ell = 0, \cdots, L$ is based on eq. (8).

In other words, initially, one samples a conditional path given by eq. (8) with $\ell = 0$. As the level, ℓ, increases, the conditional path is morphed to the path described by the original dynamics of eq. (5). The choice of the modified drift, $b(\mathbf{x}_t)$ facilitates the conditional path sampling and therefore the MCMC step in the Algorithm 1. In other words, the samples from the ℓ^{th} level are used as the initial condition at the $(\ell+1)^{th}$ level. In this way, at the final level, one samples from the original stationary density with a better initial condition, which was obtained through the preceding levels. Intuitively, the drift homotopy technique provides the MCMC step in Algorithm 1 a better initial condition.

The studies in [23], [17], [18] considered a fixed number of L levels which were employed. However, in most cases, the MCMC may achieve a convergent result prior to going through all the auxiliary levels of drift homotopy. In other words, computational time is unnecessarily spent. However, if one considers a surveillance distributed sensor network [27] which employs a particle filter method for monitoring threats, then it is of paramount importance to execute quickly and accurately the procedure since there exist stringent power constraints. In other words, a technique, which decreases the number of levels such that computational time is saved, is urgently needed. Therefore, we introduce a learning method within the MCMC sampler in the particle filter. The learning method automatically adjusts the number of levels, ℓ_k, at a given time k.

The learning criterion is the effective sample size (ESS). The ESS is a measure of how much the samples at any given time k contribute to the approximation of the filtering distribution of eq. (1). The novel learning drift homotopy particle filter calculates the ESS after each level of drift homotopy at each time step when observations are available. Suppose that one generates N i.i.d samples from the importance distribution $q(\mathbf{x})$, then ESS is defined,

$$\mathrm{ESS}_\ell = \frac{N}{1+\mathrm{CV}_{N,\ell}^2},$$

where $\mathrm{CV}_{N,\ell}$ is the coefficient of variation of the normalized weights given by

$$\mathrm{CV}_{N,\ell} = \left(\frac{1}{N} \sum_{i=1}^{N} \left(\frac{N w_i^\ell}{\sum_{j=1}^{N} w_j^\ell} - 1 \right)^2 \right)^{1/2},$$

where w_i^ℓ and w_j^ℓ denote the importance weights for the i^{th} and j^{th} particles respectively after ℓ^{th} level of drift homotopy. Notice the weights w_i^ℓ and w_j^ℓ are still calculated using (4) by substituting $\mathbf{x}_k^i$ by $\mathbf{x}_{k,\ell_k}^i$, where $\mathbf{x}_{k,\ell_k}^i$ denotes the i^{th} sample at ℓ_k^{th} level at time step k. Based on the definition of ESS, its value is between 1 and N. If the particles with equal weights $\frac{1}{N}$ are considered, then the $\mathrm{CV}_{N,\ell}$ will be equal to zero. On the other hand, if all the normalized weights but one are null, then the $\mathrm{CV}_{N,\ell}^2$ will reach its maximum value $N-1$ and therefore the ESS will be just 1. Also, the ESS reveals that using N weighed samples generated from the importance density to approximate the filtering distribution is equivalent to using $\frac{N}{1+\mathrm{CV}_{N,\ell}^2}$ i.i.d samples drawn from the filtering distribution [8], [21].

The novel learning drift homotopy particle filter employs the ESS at each time step when observations are available. Since ESS indicates the number of samples that essentially contributes to the estimation, if the ESS exceeds an appropriate threshold, it implies that the MCMC step converged to the filtering distribution, and therefore no more levels in the drift homotopy are needed. Algorithm 2 presents a pseudocode of the learning drift homotopy particle filter.

Remark 3.1: Daum and Huang [10] also introduced a homotopy technique different from herein. Their method considers an appropriate ordinary differential equation (ODE) to implement the Bayes rule rather than the pointwise multiplication of two functions, i.e. the prior and likelihood. Moreover, the authors applied the homotopy technique at the densities level whereas our scheme uses the homotopy in the dynamics.

IV. Numerical results

In this section, we present two examples to illustrate the advantages of the learning drift homotopy algorithm. The first numerical experiment is performed for a problem of a partially observed diffusion in a double well potential with small noise and the second is a multi-target tracking problem. The goal is to show how the learning drift homotopy technique helps to design an efficient particle filter.

Initialization at time $k = 0$;
for $k = 1 : T$ **do**
  Sample N unweighted samples $\mathbf{x}^i_{0:k-1}$ from $p(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k-1})$;
  Prediction: Generate N samples, $\tilde{\mathbf{x}}^i_k$ from $p(\mathbf{x}_k|\mathbf{x}^i_{k-1})$ and set $\tilde{\mathbf{x}}^i_{0:k} = (\mathbf{x}^i_{0:k-1}, \tilde{\mathbf{x}}^i_k)$;
  Update: Compute the weights by (4);
  Resampling: Generate N independent uniform random variables $\{\theta^i\}_{i=1}^N$ from $(0, 1)$. For $i, j = 1, \ldots, N$, let $\mathbf{x}^i_{0:k} = \tilde{\mathbf{x}}^j_{0:k}$ where

$$\sum_{l=1}^{j-1} w^l_k \leq \theta^j < \sum_{l=1}^{j} w^l_k$$

  MCMC step: Choose suitably modified drift and use drift homotopy to construct a Markov chain for $\mathbf{z}^{i,\ell_k}_{0:k}$ with initial value $\mathbf{x}^i_{0:k}$. At $\ell_k = 0$ begin with a sample from the modified drift of eq. (8);
  Sample through MCMC the density (9);
  while $\ell_k = 1, \ldots$ **do**
    take the last sample from the $(\ell_k - 1)^{th}$ level and use it as an initial condition for MCMC sampling of the density at the next ℓ^{th}_k level;
    if *ESS at level* $\ell_k >$ *threshold* **then**
      use the original dynamics with initial condition the samples at the level ℓ_k
    end
  end
end

Algorithm 2: Learning Drift Homotopy Particle Filter

A. *Double well potential*

Consider a diffusion in a double well potential in discrete time given by

$$\mathbf{x}_{t+\Delta} = \mathbf{x}_t - 4\mathbf{x}_t(\mathbf{x}_t^2 - 1)\Delta + \sigma\sqrt{\Delta}\xi_t, \qquad (10)$$

where the sampling rate is $\Delta = 0.01$, $\sigma = \frac{1}{2}$ and ξ_t is a standard normal random variable.

The drift of the dynamics describes a gradient flow for the potential $U(\mathbf{x}) = \mathbf{x}^4 - 2\mathbf{x}^2$ which has two equilibrium states at $\mathbf{x} = \pm 1$. Without the stochastic term, the solution will only meander in the vicinity of one of the two equilibria. A weak stochastic term as in eq. (10) will jitter the solution to walk between the two equilibria but with rather low probability.

We set the observations to be small perturbations of the equilibria points ± 1, i.e. $\mathbf{y}_k = \pm 1 + \eta_k$, where $\eta_k \sim \mathcal{N}(0, 0.04)$. Moreover, the observations are collected at every $1s$ and for $T = 20s$. We observe that the sampling time of the process is much faster $(\Delta = 0.01s)$ than the data collection $(1s)$. This setting is difficult for the particle filter to follow due to rare transitions. However, we employ here the learning drift homotopy particle filter with a modified drift $b(\mathbf{x}_t) = -c\mathbf{x}_t(\mathbf{x}_t^2 - 1)$, where $c = 0.4$ in our experiment. The modified drift corresponds to a double well potential with much shallower wells. This choice of modified drift will ease the transitions between the two equilibria.

We employ the Metropolis-Hasting algorithm as the MCMC sampler at each level $\ell_k = 0, \cdots, L$ of each time step k. The acceptance rate is given in the following

$$\alpha_{\ell_k} = min\{1, \frac{J_{\ell_k}(\mathbf{x}_k|\mathbf{x}'_k)p^{\ell_k}(\mathbf{x}'_k|\mathbf{x}_{k-1})p(\mathbf{y}_k|\mathbf{x}'_k)}{J_{\ell_k}(\mathbf{x}'_k|\mathbf{x}_k)p^{\ell_k}(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{y}_k|\mathbf{x}_k)}\}$$

where $\mathbf{x}_k$ is the current state and $\mathbf{x}'_k$ is the proposed state generated through the proposal distribution $J_{\ell_k}(\cdot)$ which has a pertinent Gaussian kernel, and $p^{\ell_k}(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{y}_k|\mathbf{x}_k)$ is the stationary distribution at level ℓ_k that can be approximated by (9).

One may fix both the number of levels, and the steps of MCMC sampling at each time step. However, this leads to unnecessary consumption of the computational time which may be disastrous if one tracks threats with a distributed surveillance sensor network. Therefore, we use the novel Algorithm 2 such that the number of levels, which are needed in order to reach a convergent result, decreases drastically. Fig.1 demonstrates the filtering estimation of the partially observed diffusion in a double well potential based on the learning drift homotopy particle filter. Also, the tracking error, which is defined as the distance between estimated state and the true state, is given in Fig.2. Moreover, a comparison of two ways of implementations is shown in Table I and Table II. Table I displays the results of a drift homotopy particle filter with fixed number of levels $(L = 40)$ and 150 MCMC steps. Table II uses the learning drift homotopy particle filter where the ESS threshold is set to be 75%. One may observe that we have comparable errors however with significantly less levels of drift homotopy and MCMC steps. In fact, there were a few instances $(t = 2, 6, 14)$ where sampling directly from the modified dynamics $(\ell = 0)$ was sufficient for the filter to reach a convergent result.

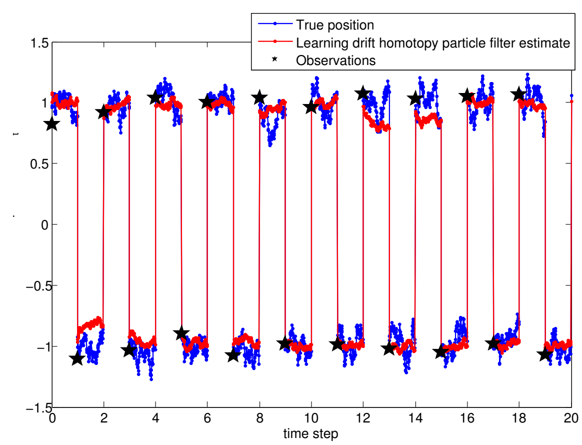

Fig. 1. In this simulation, we use learning drift homotopy particle filter with 10 samples for 20 time steps for the double well potential diffusion.

B. *Multi-target tracking*

We next study the performance of learning drift homotopy particle filter for a multi-target tracking problem. We consider two cases: a linear Gaussian model and a nonlinear non-Gaussian case.

Time steps	L	MCMC steps	Error
1	40	150	0.016962
2	40	150	0.058091
3	40	150	0.026314
4	40	150	0.046993
5	40	150	0.032146
6	40	150	0.012817
7	40	150	0.012912
8	40	150	0.034872
9	40	150	0.044780
10	40	150	0.026893
11	40	150	0.029236
12	40	150	0.015401
13	40	150	0.071464
14	40	150	0.036774
15	40	150	0.038505
16	40	150	0.039479
17	40	150	0.071885
18	40	150	0.064279
19	40	150	0.000960
20	40	150	0.002162

TABLE I. THIS TABLE SHOWS, FOR FIXED 40 LEVELS OF DRIFT HOMOTOPY AND 150 MCMC STEPS, THE ERROR AT EACH TIME STEP WHICH IS SIMPLY THE DIFFERENCE BETWEEN TRUE STATE AND ESTIMATION. THE NUMBER OF PARTICLES IS CONSIDERED TO BE 10.

Time steps	ℓ_k	MCMC steps	Error
1	22	10	0.049871
2	0	1	0.046551
3	1	29	0.048307
4	2	49	0.044896
5	7	83	0.049962
6	0	12	0.004473
7	1	37	0.049107
8	6	57	0.045660
9	24	5	0.048551
10	5	97	0.047620
11	15	15	0.043014
12	9	92	0.046975
13	9	58	0.048404
14	0	45	0.049052
15	3	29	0.047892
16	11	82	0.046791
17	5	70	0.046585
18	6	15	0.041894
19	10	35	0.049664
20	6	44	0.047678

TABLE II. THIS TABLE SHOWS THE LEVELS PERFORMED BEFORE THE FINAL LEVEL $l = 40$ AND THE MCMC STEPS, AND THE ERROR AT EACH TIME STEP WHICH IS SIMPLY THE DIFFERENCE BETWEEN TRUE STATE AND ESTIMATION. THE SAMPLE SIZE IS SET TO BE 10. THE ESS THRESHOLD IS SET TO BE 75% OF THE SAMPLE SIZE.

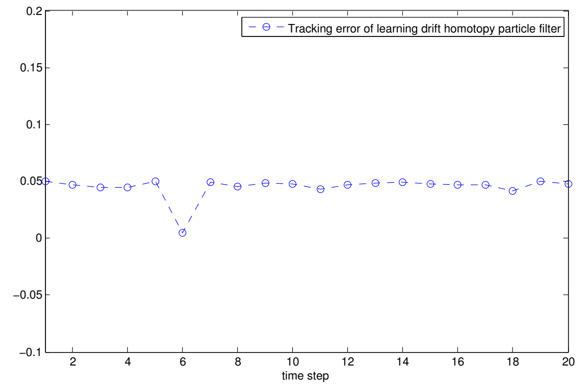

Fig. 2. Tracking error of the LDHPF shown in Fig.1

1) Case 1: Linear Gaussian model: Suppose there are m targets and the state vector of the m^{th} target at time k is represented via $\mathbf{x}_k^m = [x_k^m, \dot{x}_k^m, y_k^m, \dot{y}_k^m]$, where $(x_k^m, \dot{x}_k^m)$ and $(y_k^m, \dot{y}_k^m)$ are the position and velocity on the x and y axes respectively. The dynamic of each target is given by

$$\mathbf{x}_k^m = \mathbf{A}_1 \mathbf{x}_{k-1}^m + \mathbf{A}_2 \mathbf{u}_k^m, \tag{11}$$

where the matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ are as follows

$$\mathbf{A}_1 = \begin{pmatrix} 1 & \Delta & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta \\ 0 & 0 & 0 & 1 \end{pmatrix}, \mathbf{A}_2 = \begin{pmatrix} \Delta^2/2 & 0 \\ \Delta & 0 \\ 0 & \Delta^2/2 \\ 0 & \Delta \end{pmatrix}$$

and $\Delta = 1$ denotes the time between observations. The noise $\mathbf{u}_k^m$ is distributed according to a 2 dimensional Gaussian distribution with mean 0 and covariance,

$$\Sigma_u^m = \begin{pmatrix} 0.7 & 0 \\ 0 & 0.7 \end{pmatrix}.$$

In our simulation, a linear observation model is considered

$$\mathbf{y}_k^n = \tilde{\mathbf{x}}_k^m + \mathbf{v}_k^n, \tag{12}$$

where $\tilde{\mathbf{x}}_k^m = (x_k^m, y_k^m)^T$ is the position of the m^{th} target at time k and $\mathbf{v}_k^n$ is a Gaussian noise with covariance

$$\Sigma_v^n = \begin{pmatrix} 0.004 & 0 \\ 0 & 0.004 \end{pmatrix}.$$

We do not have prior knowledge of target-to-observation association and therefore we use the Munkres algorithm [7] to match the observations with targets.

2) Case 2: Nonlinear non-Gaussian model: In this numerical experiment, a nonlinear non-Gaussian observation model is considered which consists of the measurements of bearing θ and the range r of a target. Let $\mathbf{y}_k^n$ be the nth observation from the m^{th} target at time k, the observation model is defined below

$$\mathbf{y}_k^n = \left(arctan(\frac{y_k^m}{x_k^m}), \sqrt{(x_k^m)^2 + (y_k^m)^2} \right)' + \mathbf{v}_k^n. \tag{13}$$

where $\mathbf{v}_k^n$, is distributed according to a suitable Gaussian Mixture Model (GMM) with probability density

$$p(\mathbf{v}_k^n) = \sum_{\ell=1}^{2} w_\ell^v \mathcal{N}(\mu_{\ell,v}^n, \Sigma_{\ell,v}^n), \tag{14}$$

where $w_1^v = 0.8$, $w_2^v = 0.2$ and $\mu_{1,v}^n = -0.01$, $\mu_{2,v}^n = 0.01$ and the covariance matrices are defined as follows

$$\Sigma_{1,v}^n = \begin{pmatrix} 0.004 & 0 \\ 0 & 0.004 \end{pmatrix}, \Sigma_{2,v}^n = \begin{pmatrix} 0.0001 & 0 \\ 0 & 0.0001 \end{pmatrix}.$$

Also in this case, the driving noise in the dynamics model is considered to be a suitable GMM with two Gaussian mixands defined as follows

$$p(\mathbf{u}_k^n) = \sum_{\ell=1}^{2} w_\ell^u \mathcal{N}(\mu_{\ell,u}^n, \Sigma_{\ell,u}^n), \tag{15}$$

The means of the two Gaussians are also ± 0.01 and the covariance matrices are given in the following

$$\Sigma_{1,u}^n = \begin{pmatrix} 0.7 & 0 \\ 0 & 0.7 \end{pmatrix}, \Sigma_{2,u}^n = \begin{pmatrix} 0.1 & 0 \\ 0 & 0.1 \end{pmatrix}.$$

with weights $w_1^u = 0.8$, $w_2^u = 0.2$ respectively.

3) Drift homotopy for multi-target tracking model: The modified dynamics of target m at level ℓ for target m are given in the following

$$\mathbf{x}_{i,k}^{\ell,m} = \mathbf{A_1}\mathbf{x}_{i,k-1}^{\ell,m} + \mathbf{A_2}u_k^m + \mathbf{A_3}^{\ell,m}, \tag{16}$$

where the subscript $i = 1, \cdots, N$ corresponds to the i^{th} particle, and N denotes the sample size used in the particle filter, and

$$\mathbf{A_3}^{\ell,m} = (1-\epsilon_\ell)\mathbf{A_2}\mathbf{M}_{i,k-1}^m,$$

where

$$\mathbf{M}_{i,k-1}^m = \frac{2}{\Delta^2}\begin{pmatrix} \bar{\mu}_x^m - x_{i,k-1}^m - 2\dot{x}_{i,k-1}^m\Delta \\ \bar{\mu}_y^m - y_{i,k-1}^m - 2\dot{y}_{i,k-1}^m\Delta \end{pmatrix},$$

$\epsilon_\ell = 1/L$, and $\ell = 0, \cdots, L$ and $\bar{\mu}_x$ and $\bar{\mu}_y$ correspond to a mean drift while at the time offsetting the individual sample's properties

$$\begin{aligned} \bar{\mu}_x^m &= \frac{1}{N}\sum_{j=1}^{N}(x_{j,k-1}^m + \dot{x}_{j,k-1}^m\Delta), \\ \bar{\mu}_y^m &= \frac{1}{N}\sum_{j=1}^{N}(y_{j,k-1}^m + \dot{y}_{j,k-1}^m\Delta). \end{aligned}$$

where $j = 1, \cdots, N$ is the index of the particle.

As shown in Fig. 3, when the number of levels increases, the ESS grows and root mean square error (RMSE) reduces. The RMSE is calculated at each time step by the following formula (17).

$$\text{RMSE}(k) = \sqrt{\frac{1}{M_k}\sum_{m=1}^{M_k} \| \mathbf{x}_k^m - E[\mathbf{x}_k^m|\mathbf{y}_1, \cdots, \mathbf{y}_k] \|^2}, \tag{17}$$

Target	ℓ_k	MCMC steps	ESS (50%)
1	7	9	7.513377
2	2	16	5.026966
3	0	41	5.016717
4	0	47	7.510589
5	0	33	5.007232
6	0	23	5.012302
7	0	1	9.986915

TABLE III. THIS TABLE SHOWS THE LEVELS PERFORMED BEFORE THE FINAL LEVEL L= 20 AT A SINGLE TIME STEP $k = 10$. THE ESS THRESHOLD IS CHOSEN TO BE 50% OF THE SAMPLE SIZE. SAMPLES SIZE IS SET TO BE 10 IN THE ALGORITHM.

where $\| \cdot \|$ is the norm of the state vector. $\mathbf{x}_k^m$ is the true state vector for the m^{th} target and $E[\mathbf{x}_k^m|\mathbf{y}_1, \cdots, \mathbf{y}_k]$ is the expectation of the filtering distribution.

However, for each target and each time step, it may be superfluous to process a fixed number of levels drift homotopy of the algorithm. In this experiment, the generalized hybrid Monte Carlo [1] is employed as the MCMC sampler. The numerical results in Table (III) show that the algorithm needs much less levels for some targets at some time steps to obtain good tracking results. Also, the table specifies that the learning drift homotopy particle filter is able to automatically choose the terminating level and does not compromise the tracking performance as shown in Fig. 4. For the nonlinear and Non-Gaussian case, the tracking result is shown in Fig.5. Given that a surveillance distributed sensor network operates under limited power constraints and a quick detection and accurate tracking of targets is needed, the the threshold of the ESS has been chosen to a lower value, precisely, 50%. However, a close examination on the RMSE comparison presented in Fig. 6 implies that the learning drift homotopy algorithm estimates accurately the states of the targets while at the same time decreases the computational time by not using all levels as in the drift homotopy. Also, its performance is by far superior in comparison to particle filtering as showing in Fig. 4.

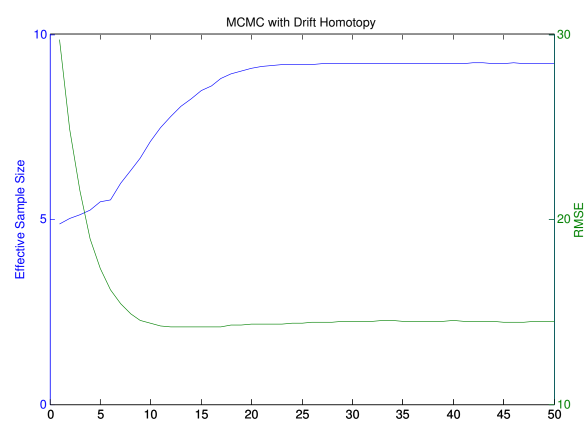

Fig. 3. This figure shows that as the level increases, the ESS increases and RMSE decreases.

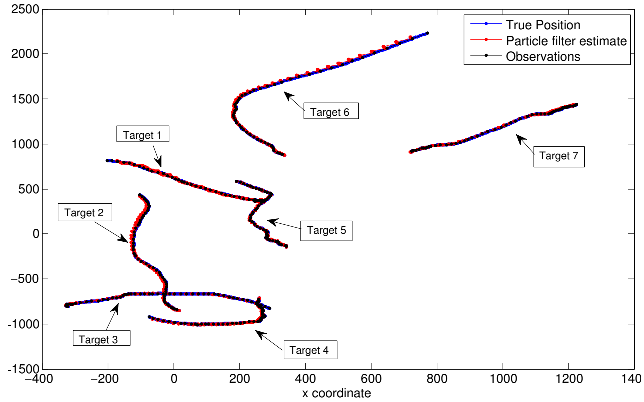

Fig. 4. Tracking result is shown in this figure with 7 targets.

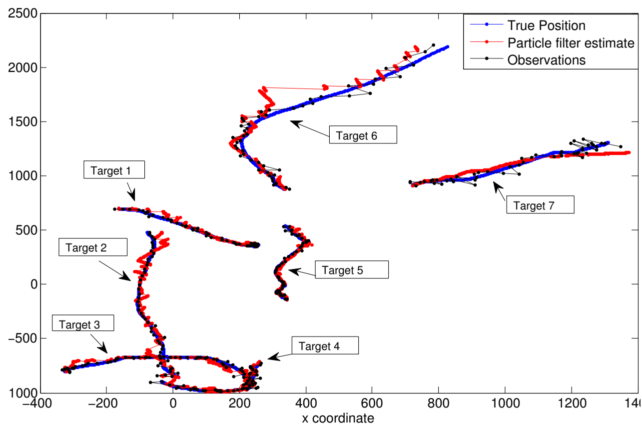

Fig. 5. Nonlinear and nonGaussian tracking result is shown in this figure with 7 targets.

V. Conclusion

In this paper, we introduced a novel learning drift homotopy particle filter algorithm. Our algorithm aims to surmount the degeneracy of particle filter by introducing a sequence of modified dynamics which facilitate the sampling. Furthermore, the algorithm "learns" from the effective sample size to adjust the number of sequences which are needed in order to reach the convergent result. This yields an accurate estimation without spending excessive computational time. Our method was successfully tested in a partially observed diffusion in a double well potential and a multi-target tracking problem. This algorithm will be fruitful when sensor nodes perform tracking in a distributed way under limited computational capabilities and stringent power constraints. [27]

Acknowledgment

Maroulas and Kang were partially supported by the Simons Foundation Award #279870. Maroulas, Kang and Schizas were partially supported by the Air Force #FA9550-15-1-0103.

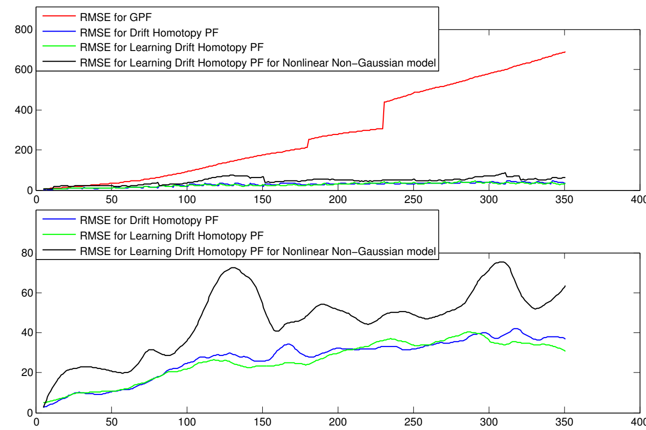

Fig. 6. This figure shows the RMSE comparison of the generic particle filter (GPF) with 500 samples, the drift homotopy particle filter and the learning drift homotopy particle filtering with 10 samples in both methods. We also include the RMSE of the learning drift homotopy particle filter for a nonlinear non-Gaussian model. The lower panel is a zoomed in and smoothed figure of the upper panel that compares the RMSE of drift homotopy particle filter with and without learning.

References

[1] F.J. Alexander, G.L. Eyink and J.M. Restrepo, Accelerated Monte Carlo for optimal estimation of time series, *Journal of Statistical Physics*, 119(516), pp. 1331-1345, June 2005.

[2] M.S. Arulampalam, S. Maskell, N. Gordon and T. Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Proc.*, 50(2), pp. 174-188, 2002.

[3] Y. Bar-Shalom and T.E. Fortmann, *Tracking and Data Association*, Academic Press, 1988.

[4] Erik P. Blasch; Modeling intent for a target tracking and identification scenario. *Proc. SPIE* Vol.5428, Signal and Data Processing of Small Targets, 2004;

[5] Erik P. Blasch; Static Versus Dynamic Data Information Fusion Analysis Using DDDAS for Cyber Security Trust. *Proc. SPIE*, Vol.29, 2014;

[6] M. Bolić, P.M. Djurić and S. Hong, Resampling algorithms for particle filters: A computational complexity perspective, *EURASIP J. on Applied Sign. Proc.,* 15, pp. 2267-2277, 2004.

[7] R.E. Burkard, M. Dell'Amico and S. Martello, *Assignment Problems*. SIAM, Philadelphia, 2009.

[8] O. Cappe, E. Moulines, and T. Ryden, *Inference in Hidden Markov Models*. Springer, 2005

[9] F. Darema, Dynamic Data Driven Applications Systems (DDDAS) ? A Transformative Paradigm *Computational Science ? ICCS 2008, Lecture Notes in Computer Science,* Volume 5103, 2008.

[10] F. Daum and J. Huang, Particle flow for nonlinear filters with log-homotopy. *Proc. SPIE 6969, Signal and Data Processing of Small Targets* 696918, 2008.

[11] A. Doucet and A. M. Johansen, *A Tutorial on particle filtering and smoothing: Fifteen years later. In Oxford Handbook of Nonlinear Filtering, D. Crisan and B. Rozovsky (eds.)*, Oxford University Press, 2011.

[12] A. Doucet, N. de Freitas and N. Gordon, *Sequential Monte Carlo Methods in Practice*, Springer NY, 2001.

[13] A. Doucet, S. Godsill and C. Andrieu, On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10, pp. 197-208, 2000.

[14] E. Evangelou and V. Maroulas, Online filtering and estimation for dynamic spatiotemporal processes with data from an exponential family, *Tech. Rep., University of Bath,* February 2015.

[15] W. Gilks and C. Berzuini, Following a moving target — Monte Carlo inference for dynamic Bayesian models, *J. Royal Stat. Soc. B*, 63(1), pp. 127-146, 2001.

[16] N.J. Gordon, D.J. Salmond and A.F.M. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation, *Proc. Inst. Elect. Eng. F*, 140(2), pp. 107-113, 1993.

[17] Kai Kang; Maroulas, V., Drift homotopy methods for a non-Gaussian filter,*Information Fusion (FUSION), 2013 16th International Conference on* , vol., no., pp.1088,1094, 9-12 July 2013

[18] Kai Kang; Maroulas, V.; Schizas, I.D., Drift homotopy particle filter for non-Gaussian multi-target tracking,*Information Fusion (FUSION), 2014 17th International Conference on*, vol., no., pp.1,7, 7-10 July 2014

[19] G. Kitagawa. Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*,5, pp. 1-25, 1996

[20] Pengpeng Liang; Teodoro, G.; Haibin Ling; Blasch, E.; Genshe Chen; Li Bai, "Multiple Kernel Learning for vehicle detection in wide area motion imagery," *Information Fusion (FUSION), 2012 15th International Conference on* , vol., no., pp.1629,1636, 9-12 July 2012

[21] J.S. Liu, *Monte Carlo Strategies in Scientific Computing*, Springer NY, 2001.

[22] R.P.S. Mahler and V. Maroulas. Tracking spawning objects. *IET Radar, Sonar & Navigation*, 7(3), pp.321-331, 2013.

[23] V. Maroulas and P. Stinis, Improved particle filters for multi-target tracking, *Journal of Computational Physics*, 231(2), pp. 602-611, 2012.

[24] Xue Mei; Haibin Ling; Yi Wu; Blasch, E.; Li Bai, Minimum error bounded efficient 1 tracker with occlusion detection,*Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* , vol.1257, no.1264, pp. 20-25 June 2011

[25] K. Murphy and S. Russell. Rao-blackwellized particle filtering for dynamic Bayesian networks, in [12].

[26] M. Pitt and N. Sheppard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94, pp. 590-599, 1999.

[27] G. Ren, V. Maroulas and I. D. Schizas. Distributed Spatio-Temporal Association and Tracking of Multiple Targets Using Multiple Sensors. *Aerospace and Electronic Systems, IEEE Transactions on* , To Appear, 2015.

[28] P. Stinis, Conditional path sampling for stochastic differential equations by drift homotopy, *Communications in Applied Mathematics and Computational Sciences*, 6(1), pp. 63-78, 2011.

[29] J. Weare, Particle filtering with path sampling and an application to a bimodal ocean current model, *Journal of Computational Physics*, 228, pp. 4312-4331, 2009.