Proceedings of the International MultiConference of Engineers and Computer Scientists 2015 Vol I,
IMECS 2015, March 18 - 20, 2015, Hong Kong

Numerical Study of a Hybrid Particle Filter

Kendra Schmal and Haiyan Cheng

***Abstract*—The particle filter method has been used in data assimilation problems for estimating states of nonlinear dynamic systems when both system errors and observation errors are present, and are possibly non-Gaussian. Weighted particles are used to represent the probability density function of the system states at any time. The particles are propagated through the system evolvement and the weights are updated. The method becomes very inefficient when the system dimension is high and the model is large and complicated. The notorious phenomenon of the method is the so called "particle degeneration" where the particles collapse until only one particle carries the majority of the weights. In this paper, we implement an improved hybrid particle filter method which aims to tackle the particle degeneration problem. The improved method introduces an auxiliary procedure which redistributes particles according to the information from the newest observation. The hybrid particle filter is tested on the Lorenz-63 model, the numerical analysis shows the method is effective for alleviating the particle degeneration problem.**

***Index Terms*—particle filter, particle swarm optimization, Lorenz 63 model**

I. Introduction

In many fields which study how various systems evolve over time, methods for performing time series analysis and time series forecasting are essential. Often such methods rely on data assimilation techniques, which use real observations in combination with a theoretical forecast model in order to reduce the uncertainty in the system prediction and improve the model forecast ability.

Data assimilation techniques optimally combine the model forecast and available observational data to generate an improved forecast. The combination process takes into consideration both the error in the model and the error in the observations. In general, there are two prevailing data assimilation techniques: sequential data assimilation and variational data assimilation. In sequential data assimilation, the assimilation occurs whenever the next observation becomes available, and observations are considered one at a time. The famous Kalman filter (KF) method produces the optimal solution. When the model is large and efficiency is a concern, the Ensemble Kalman filter (EnKF) can be used to approximate the error covariance through a Monte Carlo sampling process. Alternatively, variational data assimilation takes into account several observations within one assimilation window, minimizing the cost function to obtain a better state for the starting state of the window. Both methods have their advantages and disadvantages in terms of ease of implementation and efficiency for large scale models.

Manuscript received Jan, 2015; revised Feb, 2015. This material is based upon work supported by the National Science Foundation under Grant No. DMS-1217073.

Kendra Schmal is a undergraduate student at the Department of Computer Science, Willamette University, Salem, Oregon 97306.

Haiyan Cheng is a professor at the Department of Computer Science, Willamette University, Salem, Oregon 97306.

A particle filter is a method that models the evolution of probability densities over time. It was originally proposed by Gordon [6] as an improvement to the Extended Kalman Filter (EKF) for solving nonlinear, non-Gaussian problems. The algorithm works by recursively resampling particles based on a weight which is recalculated at each time step according to the prior probability density. In the classic implementation, the weight update is carried out through the "importance sampling" step. The method suffers from the so called "particle degeneration". In this paper, we apply an improved particle filter with the goal of alleviating the effect of the particle degeneration. Section II introduces the particle filter method and illustrates the degeneration problem. Section III describes the particle swarm optimization algorithm and its use as an auxiliary procedure to improve the particle filter performance. Numerical simulation and analysis are provided in section IV. Conclusions and future work are given in Section V.

II. Background

Forecasting problems predict system states based on the current system states and the evolvement of the system over time. The prediction process is complicated by either a large number of interrelated variables acting on the state in question or by the prevalence of randomness within the system.

The particle filter has been successfully applied to wide variety of fields, and is often used for object tracking applications and in various forecasting models, such as object tracking and robotic localization [9], [12], image and video processing [1], [10], [13], [16], as well as forecast models of natural systems, wave modeling [8], [15] and weather forecasting [7], [11].

The general non-linear system and observation equations can be represented as:

$$\begin{aligned} \theta_t &= f\left(t, \theta_{t-1}, \epsilon_{1,t}\right) \\ \xi_t &= g\left(t, \theta_t, \epsilon_{2,t}\right) \end{aligned} \tag{1}$$

in which the system states θ_t evolve according to the non-linear function f, where the function g acts as the mapping function between the state space and the observation space. $\epsilon_{1,t}$ and $\epsilon_{2,t}$ indicate the errors in system evolvement and the errors in the observations.

In a particle filter implementation, the system state is represented as a probability density function, which is approximated by an ensemble of random variables, called particles. F_t^{ξ} denotes the history of observations up to time t. With some additional assumptions, the particle filter delivers a recursive formula for the conditional density $p\left(\theta_t \middle| F_t^{\xi}\right)$ of the unobservable state θ_t, given the history of observations

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

F_t^{ξ}. Suppose we know the density $p\left(\theta_{t-1}\middle|F_{t-1}^{\xi}\right)$, and that we can sample, or generate, random variables from this density. By Bayes' formula

$$p\left(\theta_{t-1}\middle|F_t^{\xi}\right) \quad \propto \quad p\left(\xi_t\middle|F_{t-1}^{\xi},\theta_{t-1}\right)p\left(\theta_{t-1}\middle|F_{t-1}^{\xi}\right) \tag{2}$$

or

$$\frac{p\left(\theta_{t-1}\middle|F_t^{\xi}\right)}{p\left(\theta_{t-1}\middle|F_{t-1}^{\xi}\right)} \quad \propto \quad p\left(\xi_t\middle|F_{t-1}^{\xi},\theta_{t-1}\right) \tag{3}$$

The right side of Equation (3) is the observation likelihood function ξ_t at time t, and the left side of Equation (3) is a ratio of an unknown density $p\left(\theta_{t-1}\middle|F_t^{\xi}\right)$ to a known density $p\left(\theta_{t-1}\middle|F_{t-1}^{\xi}\right)$. Applying the *importance sampling* method [2], [4] to the ratio on the left side of Equation (3), we obtain an ensemble of particles drawn from the density $p\left(\theta_{t-1}\middle|F_t^{\xi}\right)$. Finally, we use this ensemble of particles and the system dynamics equation to generate an ensemble of particles drawn from the conditional density $p\left(\theta_t\middle|F_t^{\xi}\right)$. Because the filter computes the entire conditional probability density function at each time step, all the moments of the probability distribution are preserved.

A. Particle Degeneration

Particle degeneration, or sample impoverishment, occurs when the use of importance sampling fails and the particles are incorrectly positioned in the state-space. More particularly, it occurs when the majority of the particles in the resultant probability distribution are "wasted" (that is, they hold a weight value of approximately zero) while the remaining weights are concentrated in a small number of high weight particles. It was noted in [5] that particle degeneration is undesirable for two reasons. The first is that preserving particles with low weights wastes computational effort, as these particles will contribute nearly nothing to the estimation; the second is that the absence of particle diversity, sometimes manifesting in having multiple copies of the same particle, causes the filter to descend into statistical chaos.

Daum and Huang [3] note that the primary cause of particle degeneration is the use of Bayes' rule in the sampling step. Bayes' rule performs a point-wise multiplication between the prior density and the proposal density in order to obtain the posterior distribution function; this works well when the proposal distribution and the sample distribution are close, but results in weight collapse when the two probability densities overlap minimally because the particles in high-probability regions are not multiplied (or rather, are multiplied by approximately zero), and only the tail regions of the two distributions are taken into account.

Since the weight calculation step (that is, the sampling step) causes particle degeneration, it stands to reason that versions of the particle filter which have been developed for the purpose of mitigating particle degeneration typically focus on improving upon the sampling step. Most rely on the addition of a resampling step, where particles in the sample distribution are guided closer to the proposal distribution in some way or another. This helps to minimize the problem of particle degeneration, and also improves upon the accuracy of the filter overall.

III. Particle Swarm Optimization and Hybrid Particle Filter

Particle Swarm Optimization (PSO) is a swarm intelligence optimization technique based on naturally occurring swarm behavior observable in nature, such as a flock of birds or a school of fish. It was originally developed by Kennedy and Eberhart. In the PSO algorithm, the ensemble of particles each has a velocity and a state value. The velocity v and the state of the particles x are updated according to the following formula:

$$\begin{aligned} v &= \omega v + c_1 r_p(p-x) + c_2 r_g(g-x) \\ x &= x + v \end{aligned} \tag{4}$$

where p and g are the current best known individual value and best global value respectively. r_p and r_g are uniformly distributed random variables. ω (inertial weight), c_1 (cognitive weight) and c_2 (social weight) are control parameters. The difficulty of the PSO algorithm lies in the selection of those constants and setting the stopping criteria. In our experiment, we use the classic setting for those parameters and we stop the iteration when the magnitude of the error decrease is within certain threshold.

The hybrid particle filter was initially proposed by Zhang [14] and aims to alleviate the problems of particle degeneration by iteratively converging the particles onto the system observation prior to the next prediction step in the particle filter. It has been successfully applied as an auxiliary procedure to the particle filter to image and robotics tracking problems, where observations on the system play a larger role on the outcome of the system than the system model, however it has yet to be applied to forecasting problems, where observations may be subject to higher degrees of error.

In this paper, we propose a hybrid particle filter method which introduces the PSO as a sub-optimal routine between the resampling and prediction steps of the particle filter and apply the algorithm to the 3-dimensional Lorenz 63 model for atmospheric convention.

The sub-optimal routine is introduced after the resampling step in the particle filter, as a way of further addressing the problem of particle degeneration. In the swarm, the probably distribution function is represented as a swarm of particles representing the current predicted state of the system. Each particle is given a social weight c_2 which determines the importance the particles movement places on the states of the other particles, and a cognitive weight c_1 which determines the importance the particles movement places on its own state. The algorithm then iteratively attempts to attract each particle closer towards the system observation. Because the algorithm can potentially be costly with respect to time, we introduce the PSO as a sub-optimal routine, which only iterates until the given criteria (in our case, minimum change in mean) is met.

Although the introduction of an auxiliary algorithm does add to the overall runtime cost of the particle filter, the additional of the sub-optimal PSO also greatly improves upon

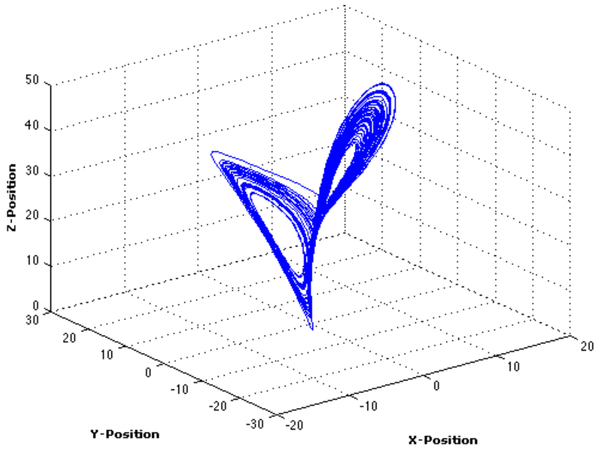

Fig. 1. Classic behavior of the Lorenz 63 Attractor

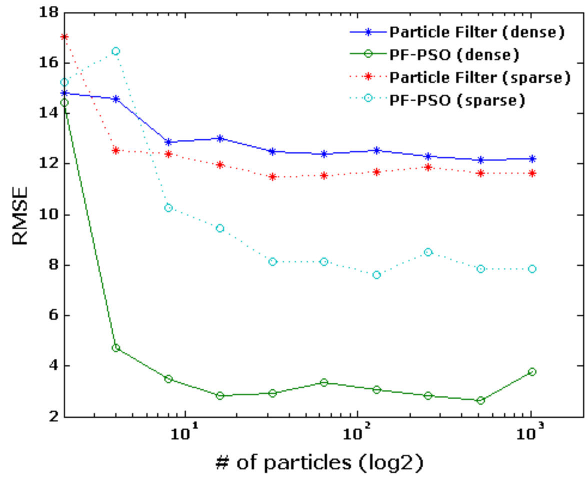

Fig. 2. Log plot of RMSE

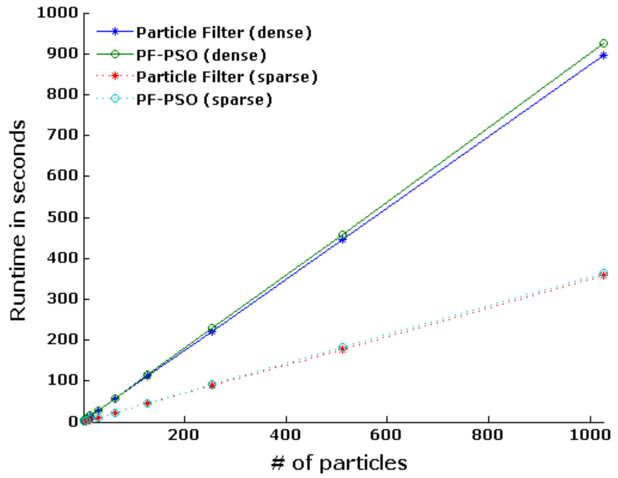

Fig. 3. Runtime Cost

the particle filters performance. The following sections shows the numerical analysis results.

IV. Numerical Simulation and Analysis

Tests of the algorithm were run over the 3-dimensional Lorenz 63 model shown in (5), which are used as a simplified model for atmospheric convection.

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y - x) \\ \frac{dy}{dt} &= x(\rho - z) - y \\ \frac{dz}{dt} &= xy - \beta z \end{aligned} \tag{5}$$

When calculated using the classic parameter values where $\rho = 28$, $\beta = 8/3$ and $\sigma = 10$, the Lorenz 63 equations produce the chaotic system seen in Figure 1. This chaotic behavior makes the Lorenz 63 attractor an ideal model on which to test the particle filter.

Tests were run on the ordinary particle filter and the hybridized PF-PSO using exponentially increasing ensemble sizes from 2^1 to 2^{10}, with each test running over the course of 20 time steps. Observations for the densely occurring observation tests were recorded twice per time step, while observations for sparsely occurring observation tests were recorded once every other time step.

Parameters for the sub-optimal PSO are as follows. Each particle was assigned a random velocity $v = N(-5, 5)$, an inertial weight $\omega = 2$, a cognitive weight $c_1 = 1$, and a social weight, $c_2 = 0.5$. Additionally, a stopping criteria $\Delta\mu_{k-1} - \mu_k < 0.1$ was applied between each iteration of the Particle Swarm in order to prevent the PSO from overshooting the targeting position.

The root mean square error is shown in Figures 2 for different number of ensemble size, where the x-axis shows the log_2 of the particle numbers. The lines with star markers indicate the traditional particle filter implementation and the lines with circles indicate the hybrid particle filter implementation. The solid lines are for dense observation case and the dashed lines are for sparse observations. From the figure, we can see that the hybrid particle filter has significantly reduced the error for both dense and sparse cases. However, increasing the number of ensemble members does not lead to further improvement.

The runtime analysis is shown in Figure 3, where both methods for dense observations consume more time than sparse observations. Comparing the runtime of the classic particle filter and the hybrid particle filter, we only see a slightly increase.

V. Conclusion

We implemented a hybrid particle filter with the attempt to improve the efficiency of the classic particle filter. An auxiliary procedure based on the particle swarm optimization method was added to the classic particle filter implementation. We tested the hybrid method on the Lorenz 63 model for both dense observation and sparse observation cases, as well as for different number of ensemble members. Numerical results indicate that the hybrid method improves the numerical accuracy with a slight runtime cost increase. Our results show potential benefits of using the hybrid particle filter for nonlinear non-Gaussian chaotic model. In the future, we plan to apply the hybrid method to a real application and further explore the optimal stopping criteria for particle swarm optimization procedure.

REFERENCES

[1] Howida A. Abd El-Halym, Imbaby I. Mahmoud, and S. E. -D. Habib. Particle Filter versus Particle Swarm Optimization for Object Tracking. *13th International Conference on Aerospace Sciences and Aviation Technology*, May 2009.

[2] M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A Tutorial on Particle Filters for Online Nonlinear / Non-Gaussian Bayesian Tracking. *IEEE Transactions in Signal Processing*, 50(2):174–188, 2002.

[3] Fred Daum and Jim Huang. Particle Degeneracy: Root Cause and Solution. *Signal Processing, Sensor Fusion, and Target Recognition*, 8050, 2011.

[4] Arnaud Doucet and Simon Godsill. Sequential Monte Carlo Methods: Final Report. Report, SAMSI, 2010.

[5] Xiaoyan Fu and Yingmin Jia. An Improvement on Resampling Algorithm of Particle Filters. *IEEE Transactions in Signal Processing*, 58(10):5414 – 5420, October 2010.

[6] NJ Gordon, DJ Salmond, and AFM Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140:107–113, 1993.

[7] Nishanth Lingala, N. Sri Namichchivaya, Nicolas Perkowski, and Hoong C. Yeong. Particle filtering in high-dimensional chaotic systems. *Chaos*, 22, 2012.

[8] Kazuyuki Nakamura, Tomoyuki Higuchi, and Naoki Hirose. Application of particle filter to identification of tsunami simulation model. *The Proceedings of SCIS & ISIS 2006*, pages 1890–1895, 2006.

[9] Ioannis M. Rekleitis. A Particle Filter Tutorial For Mobile Robot Localization. *Technical Report TR-CIM-04-02*, 2004.

[10] Yufei Tang, Siyao Fu, Bo Tang, and Haibo He. A modified pso based particle filter algorithm for object tracking. *Proc. SPIE*, 8750, April 2013.

[11] Peter Jan van Leeuwen. Efficient nonlinear data assimilation for oceanic models of intermediate complexity. *Statistical Signal Procession Workshop*, pages 345 – 348, June 2011.

[12] Nikos Vlassis, Bas Terwijn, and Ben Krose. Auxiliary Particle Filter Robot Localization from High-Dimensional Sensor Observations. *in Proceedings of the IEEE International Conference on Robotics and Automation*, 2002.

[13] Miaohui Zhang, Ming Xin, and Jie Yang. Adaptive multi-feature tracking in particle swarm optimization based particle filter framework. *Journal of Systems Engineering and Electronics*, 23(5):775–783, October 2012.

[14] Xiaoqin Zhang, Weiming He, and Steve Maybank. A Smarter Particle Filter. *Lecture Notes in Computer Science*, 5995:236–246, 2010.

[15] Zhixu Zhang, Chi-Wai Li, Yiquan Qi, and Yok-Sheung Li. Incorporation of artificial neural networks and data assimilation techniques into a third-generation wind-wave model for wave forecasting. *Journal of Hydroinformatics*, 8(1):65 – 76, 2006.

[16] Guangyu Zhu, Dawei Liang, Yang Liu, Qingming Huang, and Wen Gao. Improving Particle Filter with Support Vector Regression for Efficient Visual Tracking. *Image Processing*, 2005.