ATLANTIS PRESS

Advances in Intelligent Systems Research (AISR), volume 160

3rd International Conference on Modelling, Simulation and Applied Mathematics (MSAM 2018)

Local-loop Particle Filter Based on Artificial Fish Algorithm of Big Data

Jian Yu[1] and Yu Yan[2]

[1]Mathematics College, Shenyang Normal University, Shenyang, China

[2]Software College, Shenyang Normal University, Shenyang, China

***Abstract*—In the paper, we proposed big data novel filtering method – Local-loop Particle Filter Based on the Artificial Fish Algorithm (LPF-AF) for nonlinear dynamic systems. Particle filtering algorithm has been widely used in solving nonlinear/non Gaussian filtering problems. The proposal distribution is the key issue of the particle filtering, which will greatly influence the performance of algorithm. In the proposed LPF-AF, the local searching of AF is used to regenerate sample particles, which can make the proposal distribution more closed to the poster distribution. There are mainly two steps in the proposed filter. In the first step of LPF-AF, extended kalman filter was used as proposal distribution to generate particles, then means and variances of the proposal distribution can be calculated. In the second step, some particles move to toward the particle with the biggest weights. The proposed LPF-AF algorithm was compared with other several filtering algorithms and the experimental results show that means and variances of LPF-AF are lower than other filtering algorithms.**

Keywords—big data; filtering algorithm; particle filtering; extended Kalman filter; artificial fish algorithm

I. INTRODUCTION

In the filtering field, Kalman filtering is the most famous filtering method solving linear gauss issues. But in the real world, most filtering problems are non-linear, and non-linear filtering has a wide range of applications in many filed such as statistical signal processing[1], finance[2], target tracking[3], et al. Extended Kalman filtering (EKF) is a important non-linear filtering method, which uses the 2-order truncation of Kalman filtering to draw up to the non-linear feature of some problems. Due to the 2-order truncation in EKF, the accuracy is low, in some high linear environment, the EKF often fails. Particle filtering (PF) is a new type of filtering method, which samples points to draw up to the poster distribution. The key issue of PF is the choice of proposal distribution. Until now, many methods, such as EKF, UKF and some mixed algorithm are used to generate proposal distribution, which greatly improve the performance of PF. Artificial Fish (AF) is a new type of swarm intelligence algorithm. In the AF, every fishes has a certain view sight range and only can get information from other fishes in this range. After defining the fitness, these fishes always try to get the better region using try-out method, which means that when a fish wants to move, it tries to move several times and choose the best destination it can get. This process can be regarded as 'local only' (compared to the well known 'social only') model with try-out action.

In this paper, we proposed big data's novel filtering method – Local-loop Particle Filter Based on the Artificial Fish Algorithm (LPF-AF) for nonlinear dynamic systems. Particle filtering algorithm has been widely used in solving nonlinear/non Gaussian filtering problems. The proposal distribution is the key issue of the particle filtering, which will greatly influence the performance of algorithm. In the proposed LPF-AF, the local searching of AF is used to regenerate sample particles, which can make the proposal distribution more closed to the poster distribution. There are mainly two steps in the proposed filter. In the first step of LPF-AF, extended filter was used as proposal distribution to generate particles, then means and variances of the proposal distribution can be calculated. In the second step, some particles move to toward the particle with the biggest weights. The proposed LPF-AF algorithm was compared with other several filtering algorithms and the experimental results show that means and variances of LPF-AF are lower than other filtering algorithms.

The remaining of the paper is organized as follows: a brief description of GPF is presented in Section 2. The details of the new PF this paper proposed – LPF-AF is presents in Section 3. In Section 4 the proposed algorithm is compared to other several different filtering algorithms, finally, we give some concluding remarks in section 5.

II. GENERIC PARTICLE FILTER AND ARTIFICIAL FISH

A. Generic Particle Filter

The problem being addressed here is an estimating problem of the state of a system as a set of observations becomes available on-line, which can be expressed as follows:

$$\begin{aligned} x_t &= f(x_{t-1}) + w_{t-1} \\ y_t &= h(u_t, x_t) + v_t \end{aligned} \tag{1}$$

$x_t \in \Re^{n_x}$, $y_t \in \Re^{n_y}$, $u_t \in \Re^{n_u}$, $v_t \in \Re^{n_v}$ are the state of the system, the output observations, the input observations, the process noise and the measurement noise. The mappings:

$f : \Re^{n_x} * \Re^{n_x} \mapsto \Re^{n_x}$ and $h : \Re^{n_x} * \Re^{n_v} \mapsto \Re^{n_y}$ represent the deterministic process and measurement models. In the filtering paradigm, the posterior distribution is updated recursively over the current state x_t given all observations $Z_t = \{z_i\}_{i=1}^{t}$ up to

and including time t. Using Monte Carlo sampling points, particle filter executes the filtering process by generating weighted sampling points of state variances recursively. Generic particle filter algorithm can be predicted as follows [6]:

Initialization

For each particle

draw the states $x_0^{(i)}$ from the prior $p(x_0)$;

End

For each loop

importance sampling step

For each particle

sample $\hat{x}_0^{(i)} \sim q(x_t \mid x_{0:t-1}^{(i)}, y_{1:t})$

End

For each particle

evaluate the importance weights up to a normalizing constant:

$$w_t^{(i)} = w_{t-1}^{(i)} \frac{p(y_t \mid x_t^{(i)}) p(\hat{x}_t^{(i)} \mid x_{t-1}^{(i)})}{q(\hat{x}_t^{(i)} \mid \hat{x}_{0:t-1}^{(i)}, y_{1:t})} \quad (3)$$

For each particle, normalize the importance weights:

$$\tilde{w}_t^{(i)} = w_{t-1}^{(i)} [\sum_{j=1}^{N} w_t^{(j)}]^{-1} \quad (4)$$

// Re-sample

Eliminate the samples with low importance weights and multiply the samples with high importance weights, to obtain N random samples $x_{0:k}^{(i)}$ approximately distributed according to $p(x_{0:k} \mid z_{1:k})$.

Assign each particle an equal weight: $w_t^j = 1/N$.

When executing particle filtering, the choice of the proposal distribution is very important. Usually, the transition prior distribution is chosen to be the proposal distribution:

$$q(x_t \mid X_{t-1}, Y_t) = p(x_t \mid x_{k-1}) \quad (5)$$

B. Artificial Fish Algorithm

Artificial Fish (AF) is a new type of swarm intelligence algorithm. In the AF, every fish has a certain view sight range and only can get information from other fishes in this range. After defining the fitness, these fishes always try to get the better region using try-out method, which means that when a fish wants to move, it tries to move several times and choose the best destination it can get. This process can be regarded as 'local only' (compared to the well known 'social only') model with try-out action. Artificial fish state is denoted by $X = (x_1, x_2, ..., x_n)$; the density of food of the current location is denoted by $Y = f(x)$. The distance of any two fishes is denoted by $d_{i,j} = \| X_i - X_j \|$. The view range of fish is Visual. And the max speed of fish is denoted by Step. There are four typical actions of fish in the AF:

- Searching food: fish always wants to move the location with higher density of food, and it sees the range of Visual, and can try some predefined time, if finally it can not get a better location, it will move randomly.
- Assembling: if there are some other fishes in Visual, and the density of fishes is smaller than some predefined threshold, the fish will move to the center of these fishes, otherwise it will conduct the action of searching food.
- Chasing: if there are some fishes in the Visual, and the density of fish A of the location with highest density of food, the fish will move to the fish A, otherwise it will conduct the action of searching food.
- Randomly moving: it is the simplest method, the fish chooses a direction, and move along this direction.

From the actions of AF, it can be clearly that, the most importance features in AF are the re-trying process and local searching – Visual. In the proposed algorithm, we will introduce these concepts into the particle filtering process.

III. THE LPF-AF ALGORITHM

A. The Local-loop Process in Particle Filtering

In the proposed algorithm, after calculating the weights of particles before the re-sampling process in the general particle filtering, the mean of weights M and the particle with biggest weight A will be calculated. For each particle with weight smaller the M, it will move toward A for a random distance. If the weight of new location is bigger than the former one, the particle will move to the new location. Each particle can move predefined times to find a better location. And if finally it can not find a better one, it will not move. This process can be depicted as follows:

Step.1: calculate the mean of weights M

Step.2: identify the weight and location of the particle with biggest weights.

Step.3:

```
For each particle
  If weightP < M
    For i=1 to threshod
      newLocation =
orginalLocation+rand*(location of M - orginalLocation);
      if weight of newLocation > weightP
        orginalLocation = newLocation;
        break;
      end if
```

end for
end for

After the Local-loop process, the weights will be normalized again as follows:

$$\tilde{w}_t^{(i)} = w_{t-1}^{(i)}[\sum_{j=1}^{N} w_t^{(j)}]^{-1} \quad (6)$$

B. The LPF-AF Algorithm

Basing on the Local-loop process and general particle filtering algorithm, the proposed LPF-AF can be depicted as follows:

Initialization
For each particle
draw the states $x_0^{(i)}$ from the prior $p(x_0)$;
End
For each loop
importance sampling step
For each particle
sample $\hat{x}_0^{(i)} \sim q(x_t \mid x_{0:t-1}^{(i)}, y_{1:t})$
End
For each particle
evaluate the importance weights up to a normalizing constant:

$$w_t^{(i)} = w_{t-1}^{(i)} \frac{p(y_t \mid x_t^{(i)}) p(\hat{x}_t^{(i)} \mid x_{t-1}^{(i)})}{q(\hat{x}_t^{(i)} \mid \hat{x}_{0:t-1}^{(i)}, y_{1:t})}$$

Calculate the mean of weights M
Identify the weight and location of the particle with biggest weights.
For each particle
If weightP < M
For i=1 to threshod
newLocation =
orginalLocation+rand*(location of M - orginalLocation);
If weight of newLocation > weightP
orginalLocation = newLocation;
break;
end if
end for
end for
For each particle, normalize the importance weights:

$$\tilde{w}_t^{(i)} = w_{t-1}^{(i)}[\sum_{j=1}^{N} w_t^{(j)}]^{-1}$$

// Re-sample

Eliminate the samples with low importance weights and multiply the samples with high importance weights, to obtain N random samples $x_{0:k}^{(i)}$ approximately distributed according to $p(x_{0:k} \mid z_{1:k})$.

Assign each particle an equal weight: $w_t^i = 1/N$.

IV. The Simulation Experiments

A. Benchmark Function

In this paper, the following benchmark function [10] was chosen to test the proposed algorithm.

Bechmark:

$$x_t = 1 + \sin(w\pi(t-1)) + \frac{1}{2}x_{t-1} + u_t$$

$$y_t = \begin{cases} \frac{1}{5}x_t^2 + v_t, & t \le 30 \\ \frac{1}{2}x_t - 2 + v_t & t > 30 \end{cases}$$

where $w_t \sim \gamma(3, 0.5)$, $v_k \sim N(0, 1e-4)$, particle number N=100, sample time T=30, whicle the results were the average of 50 times of experiments. $R = 1e-4$, $Q = 0.75$; $\alpha = 1$, $\beta = 0$, $k = 2$, $threshold = 3$. All experiments were tested in MATLAB7.0.

In this paper, the proposed algorithm was compared to other several filtering algorithm – EKF, UKF, GPF, EKFPF, , UPF, Experimental settings are shown in the table 1.

TABLE I. RESULTS ON BENCHMARK 1

	MSE		MeanRunTime
	mean	variance	
EKF	0.6075	0.1138	-
UKF	0.2644	0.1297	-
PF	0.1601	0.1047	0.4034
PF-EKF	0.1126	0.2157	5.1792
PF-UKF	0.10339	0.1207	6.1134
LPF-AF	0.02218	0.03459	7.2126

The settings of other six filtering algorithms are the same as [4-8].

B. Experimental Results and Some Remarks

Experimental results are shown in figure 1~figure 2 and table.1. All results are the means of 100 runs, as the results of sa are close and far different with others, a enlargement fig was drawn as figure 3.

As shown in the experimental results, it is clear that, EKF has the worst mean and var results, but has the best running time. While the proposed algorithm has the best mean and var result, but has longer running time.

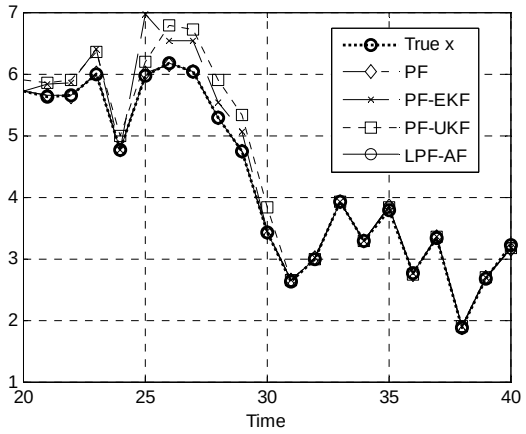

FIGURE I. RESULTS ON BENCHMARK

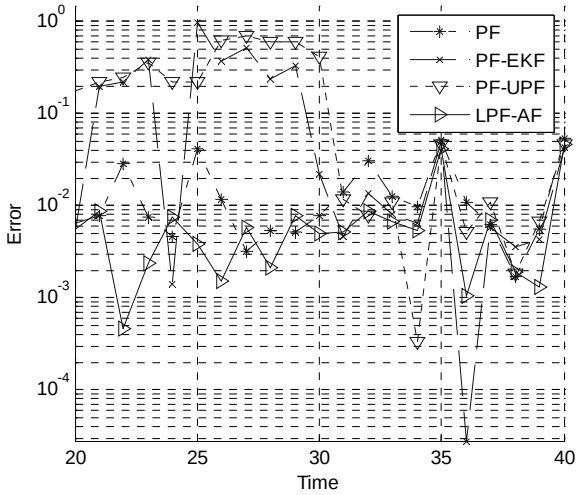

FIGURE II. RESULTS OF ERRORS ON BECHMARK

V. CONCLUSION

In this paper, we proposed big data novel filtering method – Local-loop Particle Filter Based on the Artificial Fish Algorithm (LPF-AF) for nonlinear dynamic systems. Particle filtering algorithm has been widely used in solving nonlinear/non Gaussian filtering problems. The proposal distribution is the key issue of the particle filtering, which will greatly influence the performance of algorithm. In the proposed LPF-AF, the local searching of AF is used to regenerate sample particles, which can make the proposal distribution more closed to the poster distribution. There are mainly two steps in the proposed filter. In the first step of LPF-AF, extended kalman filter was used as proposal distribution to generate particles, then means and variances of the proposal distribution can be calculated. In the second step, some particles move to toward the particle with the biggest weights. The proposed LPF-AF algorithm was compared with other several filtering algorithms and the experimental results show that means and variances of LPF-AF are lower than other filtering algorithms, but the running time is higher than others. It is clear that a suitable proposal distribution will improve the performance of particle filtering algorithm greatly. But as shown in the simulation results, the performance on the mean and var error results were improved greatly, so this cost is deserved.

REFERENCES

[1] Der Merwe R V, Doucet A. The Unscented Particle Filter[DB/OL]. Http://cslu.cse.ogi.edu/publications/ps/UPF_CSLU_talk.pdf..

[2] Riget.J. A Diversity-guided Particle Swarm Optimizer-the ARPSO. EVALife Technical Report 2012-02, Dept. of Computer Science, University of Arhus, 2012:

[3] D.Guo, Wang X, and Chen R., New sequential monte carlo methods for nonlinear dynamic systems, Statistics and Computing, vol. 15, no2, pp, 135-147, 2016.

[4] O. Cappe, S.J.Godsill, and E.Moulines, An overview of existing methods and recent advances in sequential monte carlo, IEEE Proceeding, vol.95, no.5, pp.899-924, 2017.

[5] D. Guo and X. Wang, Quasi-monte Carlo Filtering in nonlinear dynamic systems. IEEE Trans. Signal Process. vol 54, no.6.pp 2087-2098, 2015.

[6] Bar-Shalom Y, Li X R, Kirubarajan T. Estimation with Applications to Tracking and Navigation: Theory, Algorithm and Software [M]. New York: Wiley, 2016.

[7] Arulampalam M S, Maskell S, Gordon N, et al. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking [J]. IEEE Trans on Signal Processing, 2012, 20(2): 174-188.

[8] Crisan D, Doucet A. A Survey of Convergence Results on Particle Filtering methods for Practitioners [J]. IEEE Trans on Signal Processing, 2014, 50(3): 736-746..

[9] Anderson, B. D. and Moore, J.B. Optimal Filtering, Prentice-Hall, New Jersey, 2013..

[10] Julier, S.J and Uhlmann, J.K. A new extension of the Kalman filter to nonlinear systems, Proc. of AeroSense: The 11th International Sysmpsium on Aerospace/Defence Sensing, Simulation and Controls, Orlando, Florida, 2016. Vol. Muti Sensor Fusion, Tracking and Resource Mangement II p.182-193

[11] Shi Y, Eberhart R.C. A Modified Particle Swarm Optimizer. In: Proceedings of the IEEE International Conference on Evolutionary Computation. Piscataway, NJ: IEEE Press, 2015, 69-73.