I. J. Communications, Network and System Sciences. 2008; 1: 1-103
Published Online February 2008 in SciRes (http://www.SRPublishing.org/journal/ijcns/).

Particle Filtering with Multi Proposal Distributions

Fasheng WANG[1,2], Qingjie ZHAO[2], Yingbo ZHANG[1], Liqun ZHANG[2]
[1]*Neusoft Institute of Information, Northeastern University, P.R.China*
[2]*Beijing Key Lab of Intelligent Information*
School of Computer Science and Technology, Beijing Institute of Technology, P.R.China
E-mail: {wangfasheng, zhangyingbo}@neusoft.edu.cn
{zhaoqj, wangfasheng, zhangliqun}@bit.edu.cn

Abstract

Particle filtering algorithm has been applied to various fields due to its capacity to handle nonlinear/non-Gaussian dynamic problems. One crucial issue in particle filtering is the selection of the proposal distribution that generates the particles. In this paper, we give a novel strategy for selecting proposal distribution. Firstly, divide-conquer strategy is used, in which the particles used are divided into several parts. Afterward, different parts of particles are drawn from different proposal distributions. People can flexibly adjust how many of the particles drawn from specific proposal distributions according to their idiographic requirements. We provide simulation results that show its efficiency and performance.

Keywords: Sequential Monte Carlo, Proposal Distribution, Multiple Proposal Distributions

1. Introduction

As is known to all, most important real world applications are nonlinear and/or non-Gaussian. Nonlinear filtering problems exist in many fields including statistical signal processing, bio-statistics, economics, and engineering such as communications, radar tracking, sonar ranging, target tracking, car positioning, robot localization, and so on [1]. There are a variety of solutions for these problems, among which the extended Kalman filter (EKF) is well known. This kind of filters is based on linearizing technique that linearize the process model and measurement model by using Taylor series expansions. However, it is likely to diverge when the linearizing approximation methods give poor representations of the nonlinear functions.

The unscented Kalman filter (UKF) is another kind of interesting solutions, which is founded on the fact that it is easier to approximate a Gaussian distribution than it is to approximate an arbitrary nonlinear function [12][15]. Unlike the EKF, the UKF does not use the approximated models of the nonlinear process and the observation, but approximates the distribution of the state random variable by using a set of samples. It is shown that the UKF can acquire more accurate estimation results than the EKF can, but might lead to notable errors for non-Gaussian distributions.

In recent years, particle filters have drawn much attention in adaptive processing of nonlinear and non-Gaussian problems. It has been proved that the performance of particle filter is much better than traditional nonlinear filtering methods, such as the EKF, UKF, etc. [2][3].

The basic idea of particle filtering algorithm is to represent the required probability density function (PDF) by a set of random samples with associated weights. In the past decades, this method has been applied with great success to a variety of nonlinear/non-Gaussian filtering problems that was raised in communication fields, such as channel equalization [4], problems in MIMO wireless communication [5][6], phase tracking [7], problems in digital communication [19] etc. In addition, it is also widely used in vision tracking [3], robot localization [8-10], positioning and navigating [11], and so on.

A key issue in particle filtering algorithm is the selection of the proposal distribution which is generally hard to design. There are many proposal distributions proposed in the literature, among which the EKF [12][13] and the UKF [15] are well-known, as well as the transition prior [16]. Using EKF and UKF as proposal distributions, we get the Extended Kalman Particle Filter (EKPF) and the Unscented Particle Filter (UPF) [12][15],

This work was supported by the National Natural Science foundation of China, Granted NO.60772063

respectively. The famous CONDENSATION algorithm uses transition prior as the proposal distribution [16]. But the EKPF can diverge for strong nonlinearity cases, while the UPF is not applicable to general non-Gaussian model and has very high time cost. The CONDENSATION algorithm does not use the latest incoming information which contains very valuable data.

In order to solve the problems that are encountered by several existing particle filters, in this paper, we give a multi-proposal-distribution based particle filter, which is based on the EKF, UKF, and the transition prior. The particles used in the experiments are firstly divided into two parts, with one part (*c* percent) drawn from the EKF and UKF, another part (1-*c* percent) drawn from the transition prior. It gives not only higher accuracy but also lower time with proper choice of *c*.

The remainder of the paper is organized as follows. In Section 2, we give a brief introduction of particle filtering algorithm. The multi-proposal-distribution based particle filter (MPD-PF) is given in Section 3. Section 4 shows the simulation results. Conclusions are drawn in Section 5.

2. Particle Filtering

We adopt the following dynamical models:

$$x_k = f_k(x_{k-1}, v_{k-1}) \quad \textbf{(1)}$$

$$z_k = h_k(x_k, u_k) \quad \textbf{(2)}$$

where $x_k \in R^{n_x}$ denotes the system state at time *k*, and $z_k \in R^{n_z}$ denotes the observations at time *k*. The functions f_k and h_k are the system transition model function and measurement model function respectively. The noise processes are v_k and u_k, the process noise and measurement noise, respectively.

According to the basic idea of particle filters, let $\{x_{0:k}^i, w_k^i\}_{i=1}^N$ denotes a random measure used to represent the posterior density function $p(x_{0:k}|z_{1:k})$. $\{x_{0:k}^i, i = 0,..., N\}$ is a set of support particles with associated weights $\{w_k^i, i=0,…,N\}$, and $x_{0:k}$ show the states up to time *k*. When *N* tends to the infinity, the posterior density function can be approximated by:

$$p(x_{0:k} \mid z_{1:k}) \approx \sum_{i=1}^{N} w_k^i \delta(x_{0:k} - x_{0:k}^i) \quad \textbf{(3)}$$

where δ(.) denotes the Dirac delta function.

Many of the particle filters rely on the principle of importance sampling [17]. Because it is usually difficult to directly draw particles from the posterior density, we can generate particles from a proposal distribution density function *q*(), which is also known as an importance function, and the weights are assigned according to:

$$w_k^i = p(x_{0:k} \mid z_{1:k}) / q(\cdot) \quad \textbf{(4)}$$

So the choice of proposal distribution *q*(.) is of great importance. Assume the particles $x_{0:k}^i$ are drawn from an importance density $q(x_{0:k}|z_{1:k})$. Considering the following factorization:

$$q(x_{0:k} \mid z_{1:k}) = q(x_k \mid x_{0:k-1}, z_{1:k}) q(x_{0:k-1} \mid z_{1:k-1}) \quad \textbf{(5)}$$

Suppose we have gotten the approximation of the posterior distribution at time *k*-1, that is, $p(x_{0:k-1}|z_{1:k-1})$ can be represented by the particle set $\{x_{0:k-1}^i, w_{k-1}^i\}_{i=1}^N$, which are drawn from $x_{0:k-1}^i \sim q(x_{0:k-1} \mid y_{0:k-1})$. The particle weights are computed by using the formula:

$$w_{k-1}^i = p(x_{0:k-1}^i \mid z_{1:k-1}) / q(x_{0:k-1}^i \mid z_{1:k-1}) \quad \textbf{(6)}$$

In the following step, we aim to obtain $\{x_{0:k}^i, w_k^i\}_{i=1}^N$ using the new observation z_k. So long as we get the particles x_k^i and augment them onto the old particle trajectory, the new trajectory can be acquired. The new particles are generated as follows:

$$x_k^i \sim q(x_k \mid x_{0:k-1}^i, z_{0:k}) \quad \textbf{(7)}$$

The recursive equation for calculating weights can be obtained using Bayesian rules:

$$w_k^i \propto w_{k-1}^i \frac{p(z_k \mid x_k^i) p(x_k^i \mid x_{k-1}^i)}{q(x_k^i \mid x_{k-1}^i, z_k)} \quad \textbf{(8)}$$

Generally, the generic particle filter uses the transition prior $p(x_k|x_{k-1})$as the importance density function [12][16]. It has been proved that the proposal distribution $q(x_k \mid x_{0:k-1}, z_{1:k}) = p(x_k \mid x_{0:k-1}, z_{1:k})$ is optimal [12].

One of the drawbacks of particle filter is degeneracy problems. After several iterations, all but one particle would probably have negligible weights. In order to solve the problem, the resampling method is introduced [17], [18]. There are several resampling methods, such as systematic resampling, residual resmapling, etc. In this paper, the residual resampling method is used for all the experiments.

The generic particle filtering algorithm can be shown as algorithm 1.

Algorithm 1: Generic Particle Filter

Step 1. Initialization. k=0

(1) FOR i=0, …, N

Draw the states x_0^i from the prior $p(x_0)$;

END FOR

Step 2. FOR k=1, 2, …

(1) FOR i=1,…,N

Draw $x_k^i \sim q(x_k^i \mid x_{k-1}^i, z_k)$;

Assign the particle a weight according to Equ.(8);

END FOR

(2) FOR i=1,…,N

Normalize the weights $w_k^i = w_k^i / \sum_{j=1}^{N} w_k^j$;

END FOR

(3) Resample

- Eliminate the samples with low importance weights and multiply the samples with high importance weights, to obtain N random sample $x_{0:k}^i$ which are approximately distributed according to $p(x_{0:k} \mid z_{1:k})$;
- FOR i=1, …, N, let w_k^i =1/N, END FOR

END FOR

Step 3. k=k+1, goto Step2 or end executing.

3. MPD-Based Particle Filter

The particles used are firstly divided into two parts – c percent and 1-c percent. The MPD-PF first uses a mixed Kalman filter, which combines the UKF and the EKF, to draw c percent the particles. Afterwards, it uses its transition prior for another part – 1-c percent. Choice of c can affect the performance of the MPD-PF greatly. But one can choose it flexibly according to practical requirement.

For the c percent particles, suppose we have obtained the estimates of the state and the corresponding covariance at time k-1, $\bar{x}_{k-1}^{(i)}$ and $\hat{P}_{k-1}^{(i)}$. At the next time step k, the UKF is firstly used to update the particles.

The sigma points used in this process are calculated using the equation (see [12]),

$$\chi_{k-1}^{(i)a} = [\bar{x}_{k-1}^{(i)a} \quad \bar{x}_{k-1}^{(i)a} \pm \sqrt{(n_a + \lambda) P_{k-1}^{(i)a}}\ ] \tag{9}$$

The particles are propagated into the future through the nonlinear models (equation (1) and (2)),

$$\chi_{k|k-1}^{(i)x} = f(\chi_{k-1}^{(i)x}, \chi_{k-1}^{(i)v}) \quad Z_{k|k-1}^{(i)} = h(\chi_{k|k-1}^{(i)x}, \chi_{k-1}^{(i)u}) \tag{10}$$

The predicted state and corresponding covariance can be obtained,

$$\bar{x}_{k|k-1_{ukf}}^{(i)} = \sum_{j=0}^{2n_a} W_j^{(m)} \chi_{j,k|k-1}^{(i)x} \tag{11}$$

$$P_{k|k-1_{ukf}}^{(i)} = \sum_{j=0}^{2n_a} W_j^{(c)} [\chi_{j,k|k-1}^{(i)x} - \bar{x}_{k|k-1_{ukf}}^{(i)}][\chi_{j,k|k-1}^{(i)x} - \bar{x}_{k|k-1_{ukf}}^{(i)}]^T \tag{12}$$

where $W_j^{(m)}$ and $W_j^{(c)}$ are weights of sigma points (see [12] and [14]), n_a=n_x+n_v+n_u. And, the predicted measurement can be computed using,

$$\bar{z}_{k|k-1_{ukf}}^{(i)} = \sum_{j=0}^{2n_a} W_j^{(m)} Z_{j,k|k-1_{ukf}}^{(i)} \tag{13}$$

If a new measurement z_k is obtained, update the predicted estimates as follow,

$$\bar{x}_{k\ ukf}^{(i)} = \bar{x}_{k|k-1_{ukf}}^{(i)} + K_k (z_k - \bar{z}_{k|k-1_{ukf}}^{(i)}) \tag{14}$$

$$\hat{P}_{k\ ukf}^{(i)} = P_{k|k-1}^{(i)} - K_k P_{\tilde{z}_k \tilde{z}_k} K_k^T \tag{15}$$

where K_k is the Kalman gain, calculated with equation $K_k = P_{x_k z_k} P_{\tilde{z}_k \tilde{z}_k}^{-1}$,

$$P_{\tilde{z}_k \tilde{z}_k} = \sum_{j=0}^{2n_a} W_j^{(c)} [Z_{j,k|k-1}^{(i)} - \bar{z}_{k|k-1}^{(i)}][Z_{j,k|k-1}^{(i)} - \bar{z}_{k|k-1}^{(i)}]^T \tag{16}$$

$$P_{\tilde{x}_k \tilde{z}_k} = \sum_{j=0}^{2n_a} W_j^{(c)} [\chi_{j,k|k-1}^{(i)} - \bar{x}_{k|k-1}^{(i)}][Z_{j,k|k-1}^{(i)} - \bar{z}_{k|k-1}^{(i)}]^T \tag{17}$$

Here, we have obtained state and corresponding covariance estimates through UKF-update process. Consequently, we use the EKF to do the update process with the pre-computed state estimate $\bar{x}_{k\ ukf}^{(i)}$ as input.

First, predict the state and covariance using the following,

$$\bar{x}_{k|k-1_{ekf}}^{(i)} = f(\bar{x}_{k-1}^{(i)}) = f(\bar{x}_{k\ ukf}^{(i)}) \tag{18}$$

$$P_{k|k-1_{ekf}}^{(i)} = F_k^{(i)} P_{k-1}^{(i)} F_k^{T(i)} + G_k^{(i)} Q_k G_k^{T(i)} \tag{19}$$

The Kalman gain can be calculated,

$$K_k = P_{k|k-1_{ekf}}^{(i)} H_k^{T(i)} [U_k^{(i)} R_k U_k^{T(i)} + H_k^{(i)} P_{k|k-1_{ekf}}^{(i)} H_k^{T(i)}]^{-1} \tag{20}$$

So, the updated state and covariance estimates at time k are,

$$\hat{P}_{k\ ekf}^{(i)} = P_{k|k-1_{ekf}}^{(i)} - K_k H_k^{(i)} P_{k|k-1_{ekf}}^{(i)} \tag{21}$$

$$\bar{x}_{k\ ekf}^{(i)} = \bar{x}_{k|k-1 ekf}^{(i)} + \hat{P}_{k\ ekf}^{(i)} H_k^{T(i)} R_k^{-1} (z_k - h(\bar{x}_{k|k-1 ekf}^{(i)})) \quad \textbf{(22)}$$

where Q is process noise covariance and R is measurement noise covariance. $F_k^{(i)}$, $G_k^{(i)}$ and $H_k^{(i)}$, $U_k^{(i)}$ are the Jacobians of the process and measurement models, respectively. The estimates $\bar{x}_{k\ ekf}^{(i)}$ and $\hat{P}_{k\ ukf}^{(i)}$ are the estimates to be computed at time k.

For the other 1-c percent of the particles, we can directly compute the state estimates using the state transition model (2),

$$\bar{x}_{k|k-1}^{(i)} = f(x_{k-1}^{(i)}) \quad \textbf{(23)}$$

The MPD-PF algorithm can be shown as in Algorithm 2.

Algorithm 2: The MPD-PF algorithm

Step 1. Initialization: k=0

(1) FOR i=1…N, draw the particles $x_0^{(i)}$ from the prior $p(x_0)$ and set:

$$\bar{x}_0^{(i)} = E(x_0^{(i)})$$
$$P_0^{(i)} = E[(x_0^{(i)} - \bar{x}_0^{(i)})(x_0^{(i)} - \bar{x}_0^{(i)})^T]$$
$$\bar{x}_0^{(i)a} = E[x_0^{(i)a}] = [(x_0^{(i)})^T, 0, 0]^T$$
$$P_0^{(i)a} = E[(x_0^{(i)a} - \bar{x}_0^{(i)a})(x_0^{(i)a} - \bar{x}_0^{(i)a})^T] = diag(P_0^{(i)} \quad Q \quad R)$$

ENDFOR

Step 2. FOR k=1,2,…

(1). FOR i=1,…,cN,

- Update the particles using the UKF, and obtain the state estimate $\bar{x}_{k\ ukf}^{(i)}$ and covariance estimate $\hat{P}_{k\ ukf}^{(i)}$.
- Using the EKF to do the update process:
  - Let $x_{k-1}^{(i)} = \bar{x}_{k\ ukf}^{(i)}$, and compute one-step-ahead estimates of the state and the covariance with the equations (18-19).
  - Compute the updated state and covariance estimates with the equations (21-22).
  - Let $\bar{x}_k^{(i)} = \bar{x}_{k\ ekf}^{(i)}, \hat{P}_k^{(i)} = \hat{P}_{k\ ukf}^{(i)}$.
- Draw $\hat{x}_k^{(i)} \sim q(x_k^{(i)} | x_{0:k-1}^{(i)}, z_{1:k}) = N(\bar{x}_k^{(i)}, \hat{P}_k^{(i)})$

//c percent particles are drawn here.

- Assign the particle a weight, w_k^i according to Equ. (8).

ENDFOR

(2). FOR i=cN+1,…,N

- Directly compute the state estimate using Equ. (23).
- Draw $\hat{x}_k^{(i)} \sim p(x_{k|k-1}^{(i)} | x_{k-1}^{(i)})$

//1-c percent particles are drawn here

- Assign the particle a weight.

(3). FOR i=1,…,N, Normalize the weights,

$$w_k^i = w_k^i / \sum_{j=1:N} w_k^j$$

ENDFOR

(4). RESAMPLE:

- 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 the posterior PDF $p(x_{0:k}|z_{1:k})$.
- FOR i=1,…,N let w_k^i =1/N. ENDFOR

Step 3. k=k+1, goto Step2 or end executing.

4. Experimental Results

In this section, we present the experimental results of the MPD-PF algorithm and compare the performance of several existing particle filtering algorithms. The system and the measurement models used in the experiment are:

$$x_k = 1 + \sin[0.04\pi(k-1)] + 0.5x_{k-1} + v_{k-1} \quad \textbf{(24)}$$

$$z_k = \begin{cases} 0.2x_k^2 + u_k & k \le 30 \\ 0.5x_k - 2 + u_k & k > 30 \end{cases} \quad \textbf{(25)}$$

where v_k denotes a Gamma random variable and $\zeta_a(3,2)$ modeling the process noise, and the measurement noise u_k is drawn from a Gaussian distribution $N(0,0.0001)$. 200 particles are used and the process is repeated 100 times for time-steps k=1,..,60. The output of the algorithm is the mean of samples set which can be computed using (22):

$$\hat{x}_k = \frac{1}{N}\sum_{j=1}^{N} x_k^j \quad \textbf{(26)}$$

The Mean Square Errors (MSE) of each run is defined as

$$MSE = \left(\frac{1}{T}\sum_{k=1}^{T}(\hat{x}_k - x_k)^2\right)^{1/2} \quad \textbf{(27)}$$

The program run on a computer with CPU: Celeron 2.66GHz and Memory: 1GB.

If c is set to be 0.3, Figure 1 shows the comparison of the estimates to the system state generated from a single run of different particle filters. It is shown that the estimates of the PF and the EKPF bias the true state very large at some time steps, but the UPF and the MPD-PF can improve the performance.

Figure 2 shows the comparison of the Root MSE of different particle filters generated over 100 runs. It is clearly shown that the MPD-PF gives the best performance compared to other particle filters, which also can be seen in Table 1. Table 2 gives the average time of the UPF and the MPD-PF spent after 100 independent runs. The UPF spent longer running-time – 26.8 seconds, while the MPD-PF spent much shorter – 17.6 seconds. So, from Table I and II, we can clearly see that the MPD-PF gives us much higher accuracy with lower time cost, when the parameter *c* is set to be 0.3.

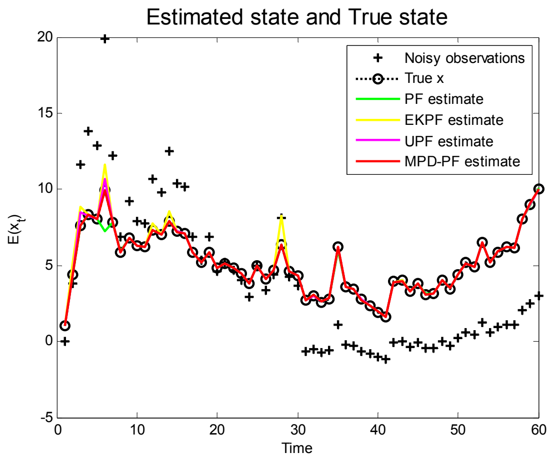

Figure 1. Estimates generated by different particle filters

Table 1. The means and the variances of the MSE over 100 independent runs (*c*=0.3).

Algorithm	MSE	
	Mean	variance
PF	0.21374	0.052091
EKPF	0.28551	0.02258
UPF	0.054599	0.004701
MPD-PF	0.016846	5.8999e-006

Figure 3 shows changing trend of the MSE of different particle filters, while the value of parameter c is increasing. From this figure, we can see that MPD-PF gives the best performance whatever the value of c. As to executing time of the particle filters, Figure 4 gives the changing trend of the executing time of the particle filters, from which we can see that, when the value of parameter c is turning bigger, the time cost of MPD-PF is increasing at the same time. At the point of *c*=0.8, it exceeds the UPF. For users with different requirements, they can flexibly adjust the value of parameter c according to their practical needs.

5. Conclusion

In this paper, multi-proposal-distribution based particle filter is introduced. It first takes a divide-conquer strategy, which draws the required particles using different proposals, and can give a better performance than several particle filters. The simulation results show that it can give much higher accuracy than the other particle filters, especially that, it can save a lot of executing time. With proper choice of *c*, the MPD-PF can supply us a fast and efficient way in dealing with some nonlinear filtering problems in real world applications, such as signal processing problems and nonlinear situations raised in wireless communications, etc. For people with different practical requirements, they can flexibly adjust the value of parameter *c* in order to obtain ideal results.

Table 2. The average time of UPF and MPD-PF used in the experiment

Algorithm	Time (seconds)
UPF	26.8
MPD-PF	16.62

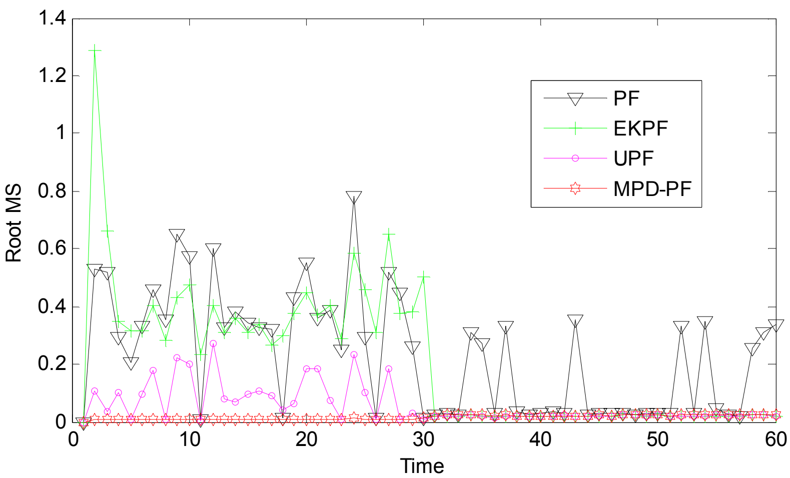

Figure 2. Root MSE of different particle filters after 100 independent runs (c=0.3).

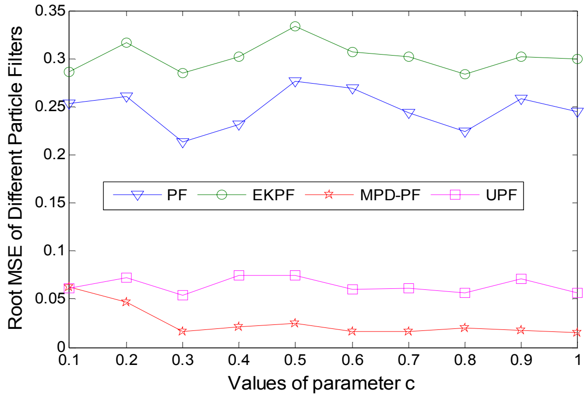

Figure 3. Changing trendline of MSE

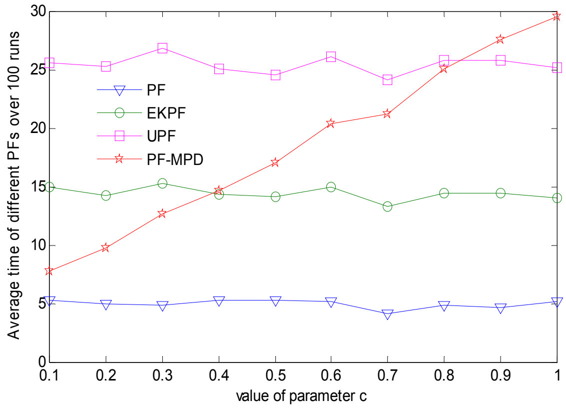

Figure 4. Changing trendline of execution time.

6. References

[1] J. H. Kotecha and P. M. Djuric, "Gaussian Particle Filtering", IEEE Transactions on signal processing. vol.51, no.10, 2003, pp. 2592-2601.

[2] N.J. Gordon, D.J. Salmond, A.F.M. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation", IEE. proceedings-F, vol.140, no.2, 1993, pp.107-113

[3] Rui Y, Chen Y., "Better proposal distributions: Object tracking using unscented particle filter", IEEE Conf. on Computer Vision and Pattern Recognition, 2001, pp. 786−793.

[4] H. Kamel, W. Badawy. "Adaptive equalization of a communication channel in a non-Gaussian noise environment", Proc. of 3rd International IEEE-NEWCAS Conference, Jun. 19-22 2005. pp.395-398

[5] Y.M. Liang, H.W. Luo, X.X. Zhao, H.B. Zhang, C.G. Y. "Nonlinear Channel Estimation Based on Particle Filtering for MIMO-OFDM Systems", Proc. of International Conference on Communications, Circuits and Systems, Vol. 1. Jun. 2006. pp.347-351.

[6] S. Haykin, K. Huber, Zhe Chen. "Bayesian Sequential State Estimation for MIMO Wireless Communications". Proc. of the IEEE. Vol. 92 Issue 3. Mar. 2004. pp.439-454.

[7] Anis Ziadi, Gerard Salut. "Non-overlapping deterministic Gaussian particles in maximum likelihood non-linear filtering- phase tracking application". Proc. of International Symposium on Intelligent Signal Processing and Communication Systems. 2005. pp. 645-648

[8] FANG Zheng, TONG Guo-Feng, XU Xin-He, "A Robust and Efficient Algorithm for Mobile Robot Localization". ACTA Automatic Sinica, Vol. 33, NO. 1. Jan. 2007. pp. 48-53.

[9] Cody Kwok, Dieter Fox, and Marina Meila, "Real-Time Particle Filters", Proceedings of the IEEE, vol.92, no. 3, Mar. 2004, pp.469-484.

[10] Christian Plagemann, Dieter Fox, Wolfram Burgard, "Efficient Failure Detection on Mobile Robots Using Particle Filters with Gaussian Process Proposals", proceedings of International Joint Conference on Artificial Intelligence, Hyderabad, India. Jan. 6-12, 2007. pp. 2185-2190.

[11] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, J. Nordlund, "Particle filters for positioning, navigation, and tracking", IEEE Transactions on Signal Processing, 2002, pp.425−437.

[12] Rudolph van der Merwe, Arnaud Doucet, Nando de Freitas, Eric Wan, "The unscented particle filter", Technical report. Cambridge University. Engineering Department. 2000.

[13] Greg Welch, Gary Bishop, "An Introduction to the Kalman Filter", Technical Report, TR 95-041, University of North Carolina at Chapel Hill, 2004.

[14] Simon J. Julier and Jeffrey K. Uhlmann, "Unscented Filtering and Nonlinear Estimation", proceedings of the IEEE, vol. 92. no. 3, 2004, pp.401-422.

[15] Eric A. Wan and Rudolph van der Merwe, "The Unscented Kalman Filter for Nonlinear Estimation", proceedings of ASSPCC, 2000, pp.153-158.

[16] Michael Isard, Andrew Blake, "Condensation – conditional density propagation for visual tracking", International Journal of Computer Vision, 1998. pp. 5~28

[17] M. Sanjeev Arulampalam, Simon Maskell, N. Gordon and T. Clapp, "A tutorial on particle filters for On-line Nonlinear/Non-Gaussian Bayesian Tracking." IEEE Transactions on signal processing, vol.50, no.2, 2002, pp.174-188.

[18] Arnaud Doucet, “On Sequential Simulation-Based Methods for Bayesian Filtering”, Technical report. Signal Processing Group, Department of Engineering, University of Cambridge, 1998.

[19] Tanya Bertozzi, Didier Le Ruyet, Gilles Rigal and Han Vu-Thien, “On Particle Filtering for Digital Communications”, Proc. of 4th IEEE Workshop on Signal Processing Advances in Wireless Communications, 2003, pp570-574.