Vol. 5(14) Jan. 2015, PP. 2030-2038

Contents list available at IJMEC

International Journal of Mechatronics, Electrical and Computer Technology (IJMEC)

Journal Homepage: www.aeuso.org

PISSN: 2411-6173 - EISSN: 2305-0543

Auxiliary Sequential Importance Resampling Particle Filter (ASIR PF) Based on Particle Swarm Optimization for Nonlinear System StateEstimation

GH. Zarei* and H. Toossian Shandiz

Department of Electrical engineering, Shahrood University of Technology, Shahrood, Iran.

***Corresponding Author's E-mail:** r.zarei_ce@shahroodut.ac.ir*

Abstract

Auxiliary Sequential Importance Resampling Particle Filter is a recursive Bayesian filtering for nonlinear systems with non-Gaussian noise which uses the Monte Carlo method for calculating the posterior probability density functions. In this filter to estimate the system state, the current observations are used to approximate the proposed distribution function and causes particles to be located in areas with a high probability. One problem with this filter and other particle filters that we are facing is the particle degeneracy. Degeneracy phenomenon increases the variance of the weight of the particles after a while thus a divergence in state estimation is created. To minimize this effect, we use Particle Swarm Optimization algorithm which directs the particles toward the greater posterior probability density functions pots.

***Keywords:** Sequential Monte Carlo, Probability density function Posterior, Degeneracy Phenomenon.*

1. Introduction

Particle Filter (PF) is a sequence Monte Carlo (SMC) for solving recursive Bayesian filtering for dealing with nonlinear and non-Gaussian systems. The basic idea of the particle filter is to use independent variables called particles to represent the past probability density function and then is to update posterior probability density function according to the new observations. This algorithm is based on Sequential Importance sampling (SIS) and the selection of proposal distribution for this is important and much research has been done in this area. In Some particulate filters Sequential Importance sampling Filter [1] and the set up particle filter [2] are considered as independent of observations for simple computing that can cause divergence. Filters developed in order to deal with this phenomenon are locally linear particle filter [3], multiple particle filters [4] and Auxiliary Sequential Importance sampling [5] proposed to maintain the distribution of observations. One of the challenges in particle filters, including Auxiliary Sequential Importance sampling filter that we face is the problem of particle degeneracy. In this phenomenon variance in weight increases after a while and it has its detrimental effects on convergence and accuracy of the filter. In practice, the occurrence of this phenomenon is a sign that, after several instances, except for one sample, all samples have been normalized weights very small. In order to reduce the variance of the particles of Particle Swarm Optimization algorithm (PSO), which is based on particle swarm intelligence, is used.

2. Particle Filter

2.1 Estimation Theory

Equations of nonlinear discrete-time state-space system are shown as follows:

Article History:
Received Date: Dec. 07, 2014
Accepted Date: Feb. 15, 2015
Available Online: Feb. 27, 2015

$$x_k = f(x_{k-1}, v_{k-1})$$
$$z_k = g(x_k, w_k) \quad (1)$$

In which $f(.,.)$ and $g(.,.)$ describe the functions of the system and observation, V_{k-1} and W_k are the system process and measurement model noise, $X_k \in R^n$ are system states and $Z_k \in R^m$ is the output of the system in Step. Eq. (1), which is the probability, can be expressed by the probability density functions of $p(Z_k \mid X_k)$ and $p(X_k \mid X_{k-1})$ as a result, state estimation and the observed values given in the following steps can be expressed by Bayes filter equations as follows:

$$p(X_{1:k} \mid Z_{1:k}) = \frac{p(Z_k \mid X_k)\, p(X_k \mid Z_{1:k-1})}{p(Z_k \mid Z_{1:k-1})} \quad (2)$$

According to the probability theory:

$$p(X_k \mid Z_{1:k-1}) = \int p(X_k \mid X_{k-1})\, p(X_{k-1} \mid Z_{1:k-1}) dX_{k-1} \quad (3)$$

Normally, it is difficult to calculate integrals due to the high dimensional state space. Hence many efforts are performed to provide approximate methods of calculating the Bayes filter. Some of these methods can be pointed out as extended Kalman filter [6], Gaussian sum filter [7] and particle filters [8]. Here we discuss a particle filter, which uses the Monte Carlo method for calculating the integral. Particle filter is a set of filters applied to the SMC method to estimate the posterior distribution [9]. The basic idea of such methods was proposed in the years before and during the 1960s and 1970s by Hammersley and colleagues ([10],[11]), and in 1993, Gordon et al article entitled “A new method for non-linear, non-Gaussian Bayesian state estimation” was published [1] that was a fundamental sequence algorithm based on Monte Carlo method to solve the problem of optimal estimation. SIS method presented in this paper was a base for most SMC filters in the past decade. This method has several names such as the bootstrap filter [1], a compression algorithm [12], particle filters [13] and interacting particle approximation [14]. The main idea of this filter is the discrete weighted approximation of the posterior probability density function by a set of $\{X_k^i, i = 1, \dots, N\}$ random samples and $\{W_k^i, i = 1, \dots, N\}$ are normalized weights. The posterior probability density function of the approximated as follows:

$$W_k^i \alpha W_{k-1}^i \frac{p(Z_k \mid X_k^i)\, p(X_k^i \mid X_{k-1}^i)}{q(X_k^i \mid X_{k-1}^i, Z_k)} \;,\; p(X_k \mid Z_k) \sim \sum_{i=1}^{N} W_k^i \delta(X_k - X_k^i) \quad (4)$$

Where $q(X_k^i \mid X_{k-1}^i, Z_k)$ is the proposed density function (or density function key), and $p(Z_k \mid X_k^i)$ is a function of the probability and $p(X_k^i \mid X_{k-1}^i)$ is transfer density function. Table 1 shows the steps of Sequential Importance sampling algorithm.

Table1: Sequential Importance sampling Algorithms

Sequential Importance sampling Algorithms(SIS)
1-Fork=0, primary particles production $\{x_0^i\}_{i=1}^{N}$ according to the probability distribution $P(x_0)$ and $w_k^i = 1/N$.
2-Fork=1,…,N, the production of particles with respect to the proposed distribution $X_k^i \sim q(X_k^i \mid X_{k-1}^i, Z_k) \;,\; W_k^i \alpha W_{k-1}^i \frac{p(Z_k \mid X_k^i)\, p(X_k^i \mid X_{k-1}^i)}{q(X_k^i \mid X_{k-1}^i, Z_k)}$
3-Normalizing the weight of the particles.

$$w_k^i = \frac{\tilde{w}_k^i}{\sum_{i=1}^{N} \tilde{w}_k^i}$$

4-State Estimation

$$\hat{x}_k^{MMSE} = \int x_k\, p(x_k | z_k) dx_k \approx \sum_{i=1}^{N} w_k^i x_k^i$$

The problem that we are facing in the SIS approach is decline in particle, and particle weight variance increases over time the amount of the variance has detrimental effects of convergence and accuracy of the filter.

2.2. Sequential Importance Resampling (SIR)

Unfortunately the SIS performance is not enough to solve the filtering problem. The main problem is related to an increase in the variance of the weights [8], after a few more time, the normalized samples will be close to zero and is just one sample of great weight. In addition to the calculation of particle weight, this will cause inappropriate estimation of the posterior distribution. For this purpose, the sample is used in the SIS filter which aims to solve the problem of degeneration by reducing the variance of the weights. During sampling, the samples are weighted at the N bottom step and samples SIS times, and the chance of being selected depends on the weight of each sample. Figure 1 shows the Resampling time. In this filter for easier calculation, a density distribution function is considered as a proposed transfer function. Table 2 shows the steps of Sequential Importance Resampling algorithm. As mentioned in this filter transmission density function is introduced as a proposed distribution function which leads to easy operation; however, if the density function of the transmission has a wider distribution than the probability function, the phenomenon of particle degeneracy may increase and to this goal, the Sequential Importance Resampling filter is used.

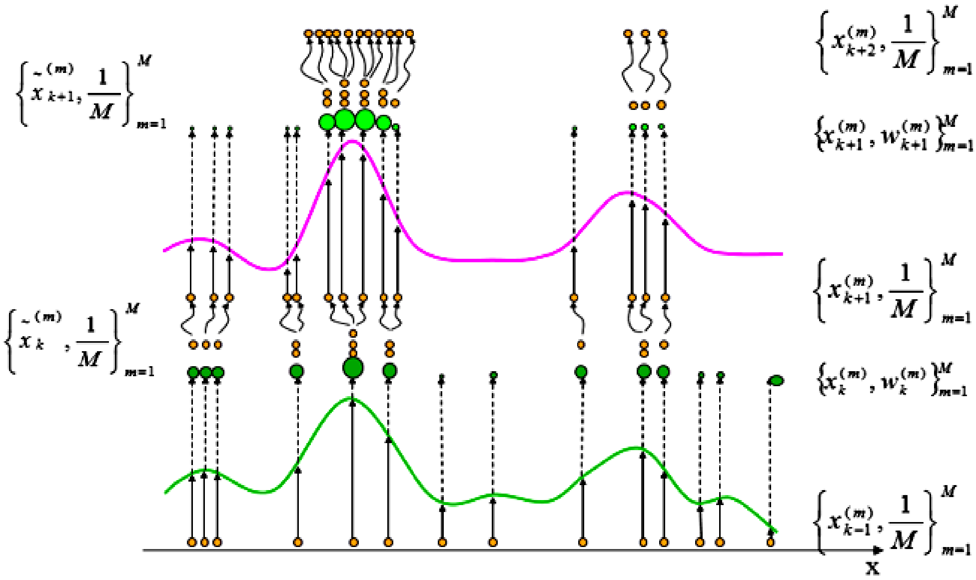

Figure 1: Resampling

Table2: Sequential Importance Resampling Algorithms

Sequential Importance Resampling Algorithms(SIR)
1- For, $K=0$, primary particles production $\{x_0^i\}_{i=1}^N$ according to the probability distribution $P(x_0)$ and $w_k^i = 1/N$
2- For, $K=1,2,\ldots$, the production of particles with respect to the proposed distribution $\tilde{w}_k^i \propto w_{k-1}^i\ p(z_k \mid x_k^i)\ x_k^i \sim q(x_k^i \mid x_{k-1}^i, z_k)$,
3-Normalizing the weight of the particles. $w_k^i = \frac{\tilde{w}_k^i}{\sum_{i=1}^N \tilde{w}_k^i}$
4- Resampling $[\{x_k^i, w_k^i, i^j\}_{i=1}^N] = \mathrm{RESAMPLE}[\{x_k^i, w_k^i\}_{i=1}^N]$
5-State Estimation $\hat{x}_k^{MMSE} = \int x_k\, p(x_k \mid z_k)\, dx_k \approx \sum_{i=1}^N w_k^i x_k^i$

2.3. Auxiliary Sequential Importance Resampling (ASIR)

ASIR filter was introduced in 1999 by Pitt and Shephard [5], as a standard SIR filter. The filter Resampling $k-1$ time is done (using observations at the k time) before the particles released at the k time. Table 3 shows the ASIR algorithm.

Table3: Auxiliary Sequential Importance Resampling Algorithms

Auxiliary Sequential Importance Resampling Algorithms(ASIR)
1- For $k=0$, primary particles production $\{x_0^i\}_{i=1}^N$ according to the probability distribution $P(x_0)$ and $w_k^i = 1/N$
2- For, $k=1,2,\ldots$, the production of particles with respect to the proposed distribution $\tilde{w}_k^i \propto w_{k-1}^i\ p(z_k \mid \mu_k^i),\ \mu_k^i \sim p(x_k \mid x_{k-1}^i)$
3-Normalizing the weight of the particles $w_k^i = \frac{\tilde{w}_k^i}{\sum_{i=1}^N \tilde{w}_k^i}$
4- sampling stage $[\{x_k^i, w_k^i, i^j\}_{i=1}^N] = \mathrm{RESAMPLE}[\{x_k^i, w_k^i\}_{i=1}^N]$
5- Production of samples obtained at k time with respect to the proposed explanation $w_k^i \propto \frac{p(z_k \mid x_k^i)}{p(x_k \mid \mu_k^{i^j})},\ x_k^i \sim q(x_k \mid i^j, z_k) = p(x_k \mid x_{k-1}^{i^j})$
6-Normalizing the weight of the particles $w_k^i = \frac{\tilde{w}_k^i}{\sum_{i=1}^N \tilde{w}_k^i}$

The ASIR filter algorithm presented in [5], consists of Sampling at the end of each iteration to establish uniform weights, but in [15] it is shown that sampling is not required in the final stage and therefore it is not part of the overall process of the algorithm which has been brought up. It is notable that if the noise of the system of equations is small, $p(X_k \mid X_{k-1}{}^i)$ is well described by$\mu_k{}^i$but if the noise of the system of equations is small, sampling in filter will be based on a bad approximation of $p(X_k \mid X_{k-1}{}^i)$ and the ASIR filter performance may be degraded.

3. Particle Swarm Optimization (PSO)

PSO algorithm was first proposed in 1995 by Eberhart and Kennedy. This algorithm, like all the other evolutionary algorithms, begins from a random population of individuals which is called here as a group of particles. Characteristics of each particle in the group are determined by a set of parameters that optimum values must be determined [16]. In this method, each particle is a point in the answer space, and each one has a memory, thus, the particle motion takes place in both directions. 1) Best position its particle failed to reach so far is $X_{i,best}$ 2) Best position ever attained by the particles is X_{gbest}. In thisway,the position of each particle in the search space is influenced by his experience and knowledge of its neighbors and itself. According to [17], the following equations show the position and velocity update of a particle:

$$v_i(t) = wv_i(t-1)+c_1r_1(x_{i,best}(t-1) - x_i(t-1)) + c_2r_2(x_{gbest}(t-1) - x_i(t-1)) \quad (5)$$

$$x_i(t) = x_i(t-1) + v_i(t) \quad (6)$$

In which the$V_i(t)$is the particle velocity in a new iteration,$V_i(t-1)$is particle velocity at the current iteration,$X_i(t-1)$is the current position of the particle,$X_i(t)$is the particle's position in the new iteration, $X_{i,best}$ is the best position the i^{th}particle has been chosen, X_{gbest} is the best position of all particles (the situation where all particles have so far provided),$r_1$and $r_2$are random numbers between 0 and 1 (which are used for diverse groups), w is the inertia weight, $c_1$and$c_2$are cognitive and social parameters respectively that selection of an appropriate value for this parameter tends to accelerate the convergence of the algorithm and avoid the convergence of early local optimization. Here,c_1,c_2 and$w$in each iteration can be considered as follows:

$$c_2(t) = \frac{2k}{K} + 0.5,\ c_1(t) = \frac{-2k}{K} + 2.5 \quad (7)$$

$$W = (Wmax - Wmin)\frac{K-k}{K} + Wmax \quad (8)$$

Where k is the current iteration and K is the total amount of iterations which are considered.
The maximum speed of the particle is considered to be as follows:

$$V_{max} = \delta(x_{max} - x_{min}) \quad (9)$$

Where X_{min} and X_{max} are minimum and maximum values of the input variables and is considered. In [18], different stages of Particle Swarm Optimization algorithm are presented.

3.1. Chaotic Particle Swarm Optimization (CPSO)

Currently, chaos as a kind of dynamic behavior of nonlinear systems has raised enormous interest in different fields of sciences such as chaos control, synchronization, pattern recognition, optimization theory and so on [19]. In random-based optimization algorithms, the methods using chaotic variables instead of random variables are called chaotic optimization algorithm (COA). Optimization algorithms based on the chaos theory are stochastic search methodologies that differ from any of the existing evolutionary computation and swarm intelligence methods. Due to the non-repetition of chaos, it can carry out overall searches at higher speeds than stochastic searches that depend on probabilities [20]. In CPSO, sequences generated by the logistic map substitute the random parameters in PSO. The parameters random are modified by the logistic map based on the following equation:

$$C_r(t+1) = \lambda \times C_r(t) \times (1 - C_r(t)) \tag{10}$$

In Eq. (10), t is current iteration in PSO algorithm, $C_r(0)$ is generated randomly for each independent run, with $C_r(0)$ not being equal to {0, 0.25, 0.5, 0.75,1} and λ equal to 4. The driving parameter λ of the logistic map, controls the behavior of $C_r(t)$ (as t goes to infinity) [21]. Since logistic maps are frequently used chaotic behavior maps and chaotic sequences can be quickly generated and easily stored, there is no need for storage of long sequences [22]. The velocity update equation for CPSO can be formulated as:

$$v_i(t) = wv_i(t-1) + c_1 C_r (x_{i,best}(t-1) - x_i(t-1)) + c_2(1 - C_r)(x_{gbest}(t-1) - x_i(t-1)) \tag{11}$$

In Eq. (11), C_r is a function based on the results of the logistic map with values between [0 1]. Fig. 2 shows the chaoticC_r value using a logistic map for 300 iterations where $C_r(0)$ = 0.001.

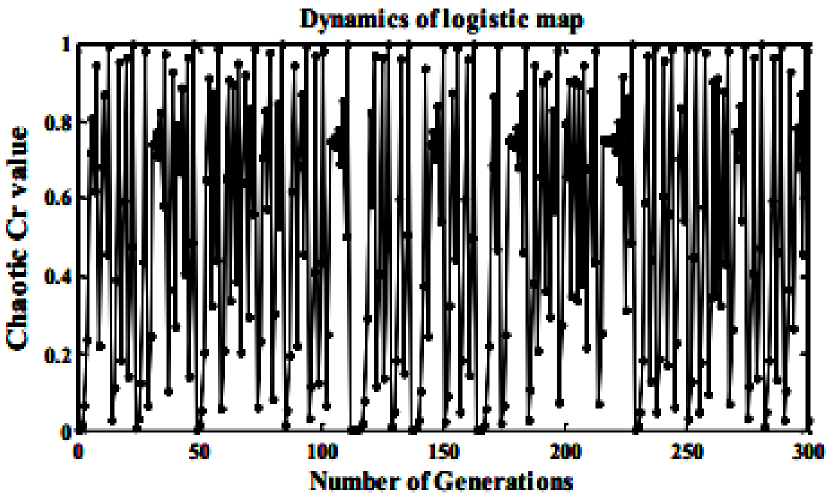

Figure 2: Chaotic $C_r$value using a logistic map for 300 iteration

4. PSO algorithm with ASIR filter optimization

The ASIR particle filters are independent of each other and do not interact with each other, but in the PSO algorithm, particles are linked together and evolve with respect to the cognitive and social effects compared to the cooperative and social aspects of animal populations and this makes PSO algorithm we use for this filter.

In optimized ASIR filter, after the particles were initialized using the prior probability density function relationship, the cost of each particle is obtained according to best positions the following equation:

$$fintess_{best\ i}(k) = e^{\left(\frac{-(z(k) - z_{best\ i}(k))^2}{2\sigma^2}\right)}, \quad fintess_i(k) = e^{\left(\frac{-(z(k) - z_i(k))^2}{2\sigma^2}\right)} \tag{12}$$

Then, using the equation of speed and position, particles in the PSO algorithm conduct to optimization and therefore particles with respect to the proposed distribution obtained are weighted. Table 4 shows the proposed algorithm.

Table4: ASIR filter based on PSO algorithm

PSO Auxiliary Sequential Importance Resampling Algorithm(PSOASIR)
1- For، $K=0$, primary particles production $\{x_0^i\}_{i=1}^{N}$ according to the probability distribution $P(x_0)$ and $w_k^i = 1/N$
2-For $K=1,2,\ldots$,the production of particles with respect to the proposed distribution $\mu_k^i \sim p(x_k \mid x_{k-1}^i)$
3-Particle Swarm Optimization Algorithm $[\{x_{best}^i\}_{i=1}^{N}] = PSO[\{x_k^i\}_{i=1}^{N}]$
4- calculating and normalizing the weight of the particles ، $\tilde{w}_k^i \propto w_{k-1}^i\, p(z_k \mid x_{best}^i)$ $w_k^i = \frac{\tilde{w}_k^i}{\sum_{i=1}^{N} \tilde{w}_k^i}$
5- sampling stage $[\{x_k^i, w_k^i, i^j\}_{i=1}^{N}] = RESAMPLE[\{x_k^i, w_k^i\}_{i=1}^{N}]$
6- Production of samples obtained at k time with respect to the proposed explanation $x_k^i \sim q(x_k \mid i^j, z_k) = p(x_k \mid x_{k-1}^{i^j})$, $w_k^i = \frac{p(Z_k \mid X_k^i)}{p(X_k \mid \mu_k^{i^j})}$
7- Normalizing the weight of the particles $w_k^i = \frac{\tilde{w}_k^i}{\sum_{i=1}^{N} \tilde{w}_k^i}$
8-State Estimation $\hat{x}_k^{MMSE} = \int x_k\, p(x_k \mid z_k)\, dx_k \approx \sum_{i=1}^{N} w_k^i x_k^i$

Benchmark model for evaluating the proposed method is the nonlinear model in [23] that:

$$x_k = f_k(x_{k-1}, k) + v_{k-1},\ f_k(x_{k-1}, k) = \frac{x_{k-1}}{2} + \frac{25x_{k-1}}{1 + x_{k-1}^2} + 8\cos(1.2(k-1)) \quad (13)$$

$$y_k = \frac{x_k^2}{20} + n_k \quad (14)$$

x_k , y_k describes the system model and measurement functions, V_{k-1} and n_k are zero mean Gaussianrandom variables with variances Q_{k-1} and R_k . In this example $Q_{k-1} = 10$, $R_k = 1$, the number of time steps 50 and 100 is the number of particles. Table 5 and Figures 2 and 3 show the results of simulation

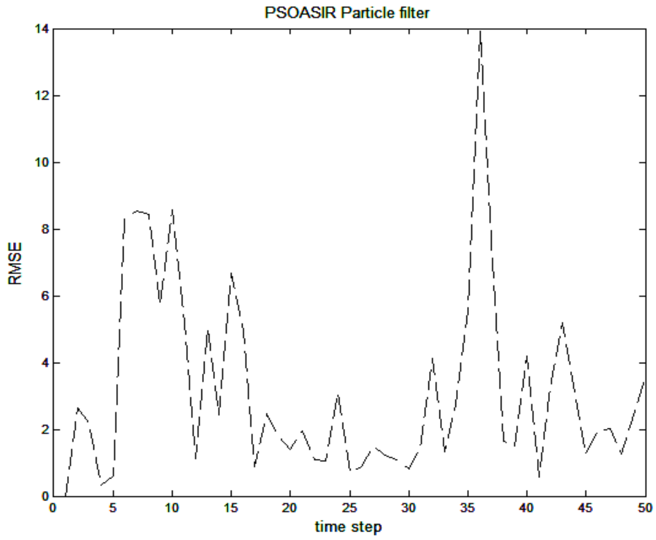

Figure 3: The estimation error of the proposed algorithm for 100 samples

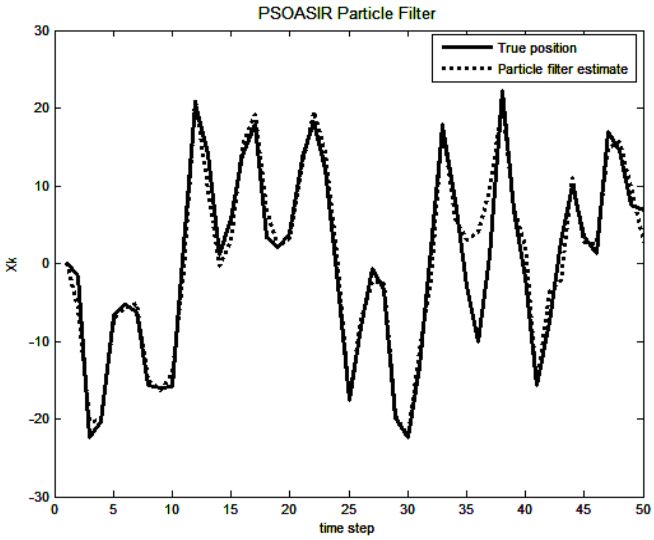

Figure4: View of the proposed algorithm estimation for 100 samples

Table5: Error estimates for the number of particles

Sample Size	Particle Filters	RMSE (m)	Time(s)
100	ASIR PF	6.06	0.214
	PSOASIR PF	5.01	0.371
	CPSOASIR PF	4.67	0.379
200	ASIR PF	5.7	0.216
	PSOASIR PF	4.4	0.527
	CPSOASIR PF	4.21	0.541
400	ASIR PF	5.5	0.22
	PSOASIR PF	4.03	0.791
	CPSOASIR PF	3.94	0.819

CONCLUSION

In this paper, we present the algorithm of PSO, to overcome the problem of the decline of ASIR particle filters. The algorithm uses current observations to estimate the prior distribution which causes the particles to move towards the maximum posterior probability density function and thus reduces the variance of the particle weights. The variance reduction makes us a better estimate of the nonlinear system.

References

[1] Gordon, N.J., Salmond, D.J., and Smith, A.F.M. *"Novel approach to nonlinear/non-Gaussian Bayesian state estimation"*. in IEE Proceedings, vol. 140, pp. 107–113, 1993.

[2] Musso, C., Oudjane, N., and LeGland, F. (2001). *"Improving regularized particle filters in Sequential Monte Carlo Methods in Practice."* (A. Doucet, N. de Freitas, and N. J. Gordon,) , New York: Springer.

[3] Doucet, A., Godsill, S., and Andrieu, C. (2000). *"On sequential Monte Carlo sampling methods for Bayesian filtering."* Statistics and Computing, Vol. 10, No. 3, pp. 197-208

[4] MacGinnity, S., and Irwin, G. W. (2000). *"Multiple model bootstrap filter for maneuvering target tracking."* IEEE Trans. on Aerospace and Electronic Systems, Vol. 36, No. 3, pp. 1006-1012.

[5] Pitt, M., and Shephard, N. (1999). *"Filtering via simulation: Auxiliary particle filters."* Journal of the American Statistical Association, Vol. 94, No. 446, pp. 590-599.

[6] Anderson, B.D.O., Moore, J.B., and Barratt, J. Optimal filtering. 1979.

[7] Alspach, D. and Sorenson, HW. *"Nonlinear bayesian estimation using gaussian sum approximations."* Automatic Control, IEEE Transactions on, 17(4):439–448, 1972.

[8] Doucet, A., Godsill, S., and Andrieu, C. *"On sequential Monte Carlo sampling methods for Bayesian filtering."* Statistics and computing, 10(3):197–208, 2000.

[9] Doucet, A., Godsill, S., and Andrieu, C. *"On sequential Monte Carlo sampling methods for Bayesian filtering. Statistics and computing"*, 10(3):197–208, 2000.

[10] J. E. Handschin and D. Q. Mayne, *"Monte Carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering"* Intern. Journal of control, vol. 9, no. 5, pp. 547-559, 1969.

[11] H. Akashi and H. kumamoto, *"Random sampling approach to state estimation in switching environments"*, Automatica, vol. 13, pp. 429-434, 1977.

[12] J. MacCormik and A. Blake, *"A probabilistic exclusion principle for tracking multiple object"*, in Proc Int. Conf. Computer Vision, pp. 572-578, 1999.

[13] J. Carpenter, P. Clifford, and P.fearnhead, *"Improved particle filter for non-linear problems"*, IEEE Proc. Part F: Radar and Sonar Navigation, vol. 146, pp. 2-7, february 1999.

[14] P. D. Maral, *"Measure valued processes and interacting particle systems: Application to non-linear filtering problems."*, Annals of Applied Probability, vol. 8, no. 2, pp. 438-495, 1998.

[15] Fearnhead, P. (1998). *"Sequential Monte Carlo Methods in Filter Theory."* PhD theris, University of Oxford.

[16] Shakerian, R., et al. *"Comparative Study of Ant Colony Optimization and Particle Swarm Optimization for Grid Scheduling [J]."* The Journal of Mathematics and Computer Science 2.3 (2011): 469-474.

[17] Moghaddas, M., et al. *"PID Control of DC motor using Particle swarm Optimization (PSO) Algorithm."* The Journal of Mathematics and Computer Science 1.4 (2010): 386-391.

[18] Zakeri, Fahimeh. *"New Particle Swarm Optimization with Diminishing Population",* The Journal of Mathematics and Computer Science 9.4 (2014): 314-320.

[19] He Y, Zhou J, Xiang X, Chen H, Qin H. *"Comparison of different chaotic maps in particle swarm optimization algorithm for long-term cascaded hydroelectric system scheduling."* Chaos SolitonsFract 2009; 42:3169–76.

[20] Coelho L, Mariani V. *"Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization."* Expert SystAppl 2008;34:1905–13.

[21] D. Kuo*, "Chaos and its computing paradigm,"* IEEE Potentials Magazine 24 (2005) 13–15.

[22] H. Gao, Y. Zhang, S. Liang, D. Li, "*A new chaotic algorithm for image encryption*," Chaos, Solitons and Fractals 29 (2006) 393–399.

[23] Van Merwe Rudolph, Doucet Arnaud.*"The unscented particle filter."* Department of Engineering, Cambridge University: Technical Report, 2000.