PSO-PF Target Tracking in Range-Based Wireless Sensor Networks with Distance-Dependent Measurement Noise

Atiyeh Keshavarz-Mohammadiyan, Hamid Khaloozadeh
Faculty of Electrical Engineering
K. N. Toosi University of Technology
Tehran, Iran

***Abstract*—In this paper a Particle Swarm Optimization (PSO) based Particle Filter (PF) for tracking a rotating object in a range-based Wireless Sensor Network (WSN) equipped with distance measuring sensors is developed. The distance-dependent measurement error is incorporated in the observation equation as a multiplicative noise. To overcome the impoverishment problem of PF, weighted aggregation of the likelihood and the prior is maximized through PSO in order to move the prior samples towards regions of the state space where both the likelihood and the prior are significant. Performance of the proposed approach is compared with that of Extended Kalman Filter (EKF) state estimator. Simulation results show the effectiveness of the developed target tracking approach.**

Keywords—Target Tracking; Wireless Sensor Networks; Particle Filter; Particle Swarm Optimization

I. INTRODUCTION

Advances in wireless technology along with Micro Electro Mechanical Systems (MEMS), has attracted many researchers to deploy Wireless Sensor Network (WSN) for target tracking applications. WSN consists of spatially distributed tiny devices called nodes which can collect measurements from the environment by their sensors and send them to a processing unit called Base Station (BS). These nodes communicate with each other over wireless links. The collected data in BS can be used for the purpose of tracking or monitoring applications.

In some WSNs referred to as range-based networks, nodes have limited range for sensing and will not be able to sense the target if their relative distance to target is more than their sensing range. Moreover, in some distance measuring sensors, the measurement noise covariance matrix grows as the distance between target and sensor increases. To deal with this problem, state dependent measurement noise is modeled as a multiplicative noise in observation equation. It would be worthy to mention that the observation equation is nonlinear when the relative distance between target and sensors is used as measurements. Since Particle Filter (PF) is able to handle the nonlinear and non-Gaussian measurement and dynamic models, it seems reasonable to use it for target state estimation while having nonlinear observation model.

PF [1]-[4] is a Monte-Carlo state estimation technique which generates a set of samples and their associated weights and updates them recursively to estimate the posterior Probability Density Function (PDF) of the state. Despite the high attention to PF, it suffers from an important disadvantage referred to as sample impoverishment. Impoverishment is a result of resampling stage of PF in which the particles with smaller weights are eliminated and the particles with high weights are propagated. Therefore, if the likelihood distribution is narrow and highly peaked compared to the prior distribution or if it lies at the tail of prior distribution, only a few particles with significant weights will be available and sample impoverishment appears.

Many researches have proposed their methods to overcome impoverishment problem. In [5], Unscented Particle Filter (UPF) has been introduced by combining Unscented Kalman Filter (UKF) [6] with PF. UPF incorporates new observation into sampling to avoid impoverishment. It uses UKF to obtain a better proposal distribution and improve the sampling step of PF. The drawback of UPF is its high computational burden.

The other approach to overcome the impoverishment problem is to incorporate some intelligent algorithms such as Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) into PF. In [7]-[8], PF has been combined with GA to improve the diversity of particle samples. In [9]-[12] PSO has been merged into PF to move the prior samples towards region of the state space with higher likelihood. In [13], a PF model which incorporates PSO has been proposed for state estimation of systems with multiplicative process and observation noises.

In this paper, a PSO based PF has been developed to overcome the impoverishment problem for target tracking in a range-based WSN in the presence of distant dependent measurement noise using the results of [13]. PSO is used before resampling step of PF to move the prior samples towards regions of the state space where both likelihood and prior are significant.

The remainder of this paper is organized as follows. In section II, the WSN architecture, target dynamics and observation model are described. Section III is devoted to describing the basics of PF. Moreover, the state transition PDF and the likelihood function for the current target tracking problem in a WSN with multiplicative measurement noise is derived in section III. In section IV, PSO is introduced. The

978-1-4799-1972-7/15/$31.00 ©2015 IEEE

target tracking approach using PSO-PF is developed in section V. Simulation experiments are performed in section VI and finally the paper is concluded in section VII.

II. Problem Formulation

In this section, we consider the problem of tracking a rotating object in a 2-D Cartesian coordinate through a range-based WSN. The WSN architecture, target motion model and observation model are described in the following.

A. WSN Architecture

The range-based WSN is composed of N_s distance measuring sensors. The sensor nodes are located at known static positions in a 2-D terrestrial field and are distributed uniformly in the area. The sensors have limited range for sensing of r_{sens} and the BS node is positioned at the center of the monitored field.

B. Target Motion Model

We assume that the target moves according to Nearly Coordination Turn (NCT) model. In the NCT model which describes the object's rotating motions, the state vector $X(k)$ is

$$X(k) = [x(k) \quad v_x(k) \quad y(k) \quad v_y(k) \quad \omega(k)]^T \tag{1}$$

whose elements are position and velocity in x coordinate and position and velocity in y coordinate and the turn rate respectively. The NCT model which characterizes the dynamic of a turn is given by

$$X(k+1) = FX(k) + Gw(k) \tag{2}$$

in which, $w(k) \sim \mathcal{N}(0, Q)$ is the zero-mean Gaussian process noise and F and G are described as follows

$$F = \begin{bmatrix} 1 & \frac{sin(\omega(k)T)}{\omega(k)} & 0 & -\frac{1-cos(\omega(k)T)}{\omega(k)} & 0 \\ 0 & cos(\omega(k)T) & 0 & -sin(\omega(k)T) & 0 \\ 0 & \frac{1-cos(\omega(k)T)}{\omega(k)} & 1 & \frac{sin(\omega(k)T)}{\omega(k)} & 0 \\ 0 & sin(\omega(k)T) & 0 & cos(\omega(k)T) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$G = \begin{bmatrix} \mathrm{T}^2/2 & 0 & 0 & 0 & 0 \\ 0 & \mathrm{T} & 0 & 0 & 0 \\ 0 & 0 & \mathrm{T}^2/2 & 0 & 0 \\ 0 & 0 & 0 & \mathrm{T} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

where, T is the sampling interval.

It should be noticed that in NCT model, the turn rate $\omega(k)$ is included in both state transition matrix and state vector. Therefore, the NCT model is nonlinear.

C. Observation Model

As mentioned before, the range-based WSN which is considered in this paper is composed of distance measuring sensors. These sensors have state dependent measurement noise which is modeled as a multiplicative noise in their observation models. Therefore, the observation model of the i^{th} sensor at the k^{th} time step is given by

$$z_i(k) = h_i\big(X(k), v(k)\big) = r_i(k)\big(1 + v_i(k)\big) \tag{4}$$
$$= \sqrt{(x(k) - x_i)^2 + (y(k) - y_i)^2}\big(1 + v_i(k)\big)$$

where, $\big(x(k), y(k)\big)$ is the target position at k^{th} time step and (x_i, y_i) is the position of i^{th} sensor. The term $v_i(k) \sim \mathcal{N}(0, R_i)$ is the zero-mean Gaussian observation noise of the i^{th} sensor and $r_i(k)$ is the relative distance between target and i^{th} sensor.

III. Particle Filter

In this section, we describe the basics of generic PF using concepts of [2]-[3]. We consider the general problem of nonlinear non-Gaussian state transition and observation models as follows.

$$X(k+1) = f\big(X(k), w(k)\big) \tag{5}$$
$$z(k) = h\big(X(k), v(k)\big)$$

where, f and h are known state transition and known observation functions and $w(k)$ and $v(k)$ are process and observation noises respectively.

To perform PF, the following steps are repeated for N_p particles in each iteration. In initialization step, the initial state sequence $\{\hat{X}^{(i)}(k|k)\}_{i=1}^{N_p}$ and their corresponding weights $\{w^{(i)}(k)\}_{i=1}^{N_p}$ are generated. The initial states can be generated using the prior PDF $\mathcal{N}\left(\hat{X}^{initial}(k|k), P^{initial}(k|k)\right)$. In sampling step, random samples $\hat{X}^{(i)}(k+1|k+1)$ are generated using the proposal distribution $q\left(\hat{X}(k+1|k+1)|\hat{X}(k|k), z(k+1)\right)$.Then, in weight calculation step, non-normalized weights are obtained

$$\widetilde{w}^{(i)}(k+1) = p\left(z(k+1)|\hat{X}^{(i)}(k+1|k+1)\right)\widetilde{w}^{(i)}(k) \tag{6}$$

and normalized

$$w^{(i)}(k+1) = \frac{\widetilde{w}^{(i)}(k+1)}{\sum_{i=1}^{N_p} \widetilde{w}^{(i)}(k+1)} \tag{7}$$

In resampling step, the number of effective samples N^{eff} is calculated

$$N^{eff} = \frac{1}{\sum_{i=1}^{N_p}\big(w^{(i)}(k+1)\big)^2} \tag{8}$$

and if N^{eff} is smaller than a predefined threshold, resampling is performed in which particles with smaller weights are eliminated and the particles with high weights are propagated. Finally, in output calculation step, the state is estimated as follows

$$\hat{X}(k+1|k+1) = \sum_{i=1}^{N_p} w^{(i)}(k+1)\,\hat{X}^{(i)}(k+1|k+1) \tag{9}$$

The choice of the proposal distribution is an important step in PF. The most common choice is using the state transition probability as the proposal distribution

$$q\left(\hat{X}(k+1|k+1)|\hat{X}(k|k), z(k+1)\right) \tag{10}$$
$$= p\big(X(k+1)|X(k)\big)$$

The state transition PDF is given by

$$p\big(X(k+1)|X(k)\big) = p_{w(k)}\big[f^{-1}\big(X(k+1), X(k)\big)\big] \tag{11}$$
$$\times \left|\det\big(\nabla_{X(k+1)}(f^{-1})^T\big)\right|$$

where, $\left|\det\left(\nabla_{X(k+1)}(f^{-1})^T\right)\right|$ denotes the absolute value of the Jacobian determinant and $p_{w(k)}(.)$ is the PDF of process noise. Similarly, the likelihood function which is used in weight calculation step is given by

$$\begin{aligned} &p\big(z(k+1)\big|X(k+1)\big) \\ &= p_{v(k+1)}\left[h^{-1}\left(z(k+1), h\left(\hat{X}(k+1)\right)\right)\right] \\ &\times \left|\det\left(\nabla_{z(k+1)}(h^{-1})^T\right)\right| \end{aligned} \tag{12}$$

where, $p_{v(k+1)}$ is the PDF of observation noise.

In section II, we described equations of state transition in NCT model by (2). As mentioned before, $w(k)\sim\mathcal{N}(0,Q)$ is a zero mean Gaussian noise. As a result $Gw(k)$ is a zero mean Gaussian noise with covariance matrix of GQG^T. Therefore, the state transition PDF is obtained as follows

$$p\big(X(k+1)\big|X(k)\big) = p_{Gw(k)}[X(k+1) - FX(k)] \tag{13}$$

The nonlinear observation model with multiplicative noise was given by (4) in section II. We can write

$$v_i(k) = \frac{z_i(k) - r_i(k)}{r_i(k)} \tag{14}$$

If we consider the observations of N_s sensors and their noises as vectors $Z(k) = [z_1(k) \quad \cdots \quad z_{N_s}(k)]^T$ and $V(k) = [v_1(k) \quad \cdots \quad v_{N_s}(k)]^T$, then we have

$$V(k) = \left[\frac{z_1(k) - r_1(k)}{r_1(k)} \quad \cdots \quad \frac{z_{N_s}(k) - r_{N_s}(k)}{r_{N_s}(k)}\right]^T \tag{15}$$

Therefore, the likelihood function $p\big(Z(k+1)\big|X(k+1)\big)$ is obtained as follows

$$\begin{aligned} p\big(Z(k+1)\big|X(k+1)\big) = p_{V(k+1)}\left(\begin{bmatrix} \frac{z_1(k+1) - r_1(k+1)}{r_1(k+1)} \\ \vdots \\ \frac{z_{N_s}(k+1) - r_{N_s}(k+1)}{r_{N_s}(k+1)} \end{bmatrix}\right) \\ \times \left|\det\left(\left(diag\left(\left[\frac{1}{r_1(k+1)}, \dots, \frac{1}{r_{N_s}(k+1)}\right]\right)\right)^T\right)\right| \end{aligned} \tag{16}$$

which results in

$$\begin{aligned} p\big(Z(k+1)\big|X(k+1)\big) = \left|\frac{1}{r_1(k+1) \times \dots \times r_{N_s}(k+1)}\right| \\ \times p_{V(k+1)}\left(\begin{bmatrix} \frac{z_1(k+1) - r_1(k+1)}{r_1(k+1)} \\ \vdots \\ \frac{z_{N_s}(k+1) - r_{N_s}(k+1)}{r_{N_s}(k+1)} \end{bmatrix}\right) \end{aligned} \tag{17}$$

where, $V(k+1)$ is a zero mean Gaussian noise with covariance matrix $R = diag([R_1, \dots, R_{N_S}])$. In (17), $r_i(k+1)$ is the estimated relative distance between the target and the i^{th} sensor which can be calculated using the state estimations in each iteration.

IV. Particle Swarm Optimization

PSO [14]-[15] is a population-based stochastic intelligent optimization technique. It consists of an objective function to be optimized and N_{pso} particles $\{x^{i,k}\}_{i=1}^{N_{PSO}}$ where k is the time step and $x^{i,k} \in R^D$ is a D-dimensional vector. The group of these particles is called the *swarm*. Each particle has an associated fitness value $f(x^{i,k})$ and a relevant velocity $v^{i,k}$. PSO is initialized with a swarm of random particles $\{x^{i,0}\}_{i=1}^{N_{PSO}}$. Each particle in PSO is moved iteratively through the problem space with its corresponding velocity. The fitness value of each particle in each iteration is the value of the objective function for that particle. The aim of PSO is to optimize this objective function. During each iteration, the fitness value of each particle is calculated. PSO has a memory in which the best position (position with best fitness value) of each particle until the current time is stored and denoted by $b^{i,k}$. The best global particle which is denoted by $gb^{i,k}$ is the particle with best fitness value amongst others until the current time. During each iteration, position of all particles are updated using the following equations

$$\begin{aligned} v^{i,k+1} = \chi[v^{i,k} + c_1u_1(b^{i,k} - x^{i,k}) \\ + c_2u_2(gb^{i,k} - x^{i,k})] \\ x^{i,k+1} = x^{i,k} + v^{i,k+1} \end{aligned} \tag{18}$$

where, constant χ is the constriction factor. c_1 and c_2 are some positive constants called cognitive acceleration and social acceleration parameters respectively, and $u_1, u_2 \in [0,1]$ are uniformly distributed random values.

The best global particle of all times and its fitness value are the solution to the optimization problem. The algorithm is ended when the fitness value reaches a predefined threshold or when a maximum number of iterations are encountered.

V. Target Tracking Approach

In this section, we describe the procedure of incorporating PSO into PF. As mentioned before, PF suffers from sample impoverishment problem which occurs if the region of state space in which the likelihood has significant values is small compared to region in which the prior has significant values. Therefore many samples will have small weights and will be eliminated in the resampling procedure. In this paper, we propose to use the introduced technique in [13] to move the prior samples towards regions of the state space where both the likelihood and the prior are significant. Therefore, we overcome the impoverishment problem for target tracking in a range-based WSN in the presence of distant dependent measurement noise. For this purpose, we add a PSO algorithm in PF before the resampling step. We use PSO to maximize the following weighted objective function so that the prior samples move towards regions with significant prior and likelihood.

$$OF = \alpha_1F_1 + \alpha_2F_2 \tag{19}$$

where, α_1 and α_2 are weighting factors anf F_1 and F_2 are prior and likelihood functions respectively and described as follows

$$\begin{aligned} F_1 = \frac{1}{\sqrt{(2\pi)^5\det(GQG^T)}} \times \\ \exp\left(-\frac{1}{2}(X(k+1)^{i,PSO} - X(k+1)^i)^T(GQG^T)^{-1}(X(k+1)^{i,PSO} \right. \\ \left. - X(k+1)^i)\right) \end{aligned} \tag{20}$$

$$
\begin{aligned}
&F_2 \\
&= \left|\frac{1}{r_1(k+1) \times \dots \times r_{N_s}(k+1)}\right| \times \frac{1}{\sqrt{(2\pi)^{N_s}\det(R)}} \\
&\times \exp\left(-\frac{1}{2}\left[\frac{z_1(k+1)-r_1(k+1)}{r_1(k+1)} \quad \cdots \quad \frac{z_{N_s}(k+1)-r_{N_s}(k+1)}{r_{N_s}(k+1)}\right]\right. \\
&\left. \times R^{-1} \times \begin{bmatrix} \dfrac{z_1(k+1)-r_1(k+1)}{r_1(k+1)} \\ \vdots \\ \dfrac{z_{N_s}(k+1)-r_{N_s}(k+1)}{r_{N_s}(k+1)} \end{bmatrix}\right)
\end{aligned}
$$

where, $X(k+1)^i, i = 1, \dots, N_p$ are the prior particle samples of PF. After performing PSO, new prior samples $X(k+1)^{i,PSO}, i = 1, \dots, N_p$ are obtained which maximize both prior and likelihood. These new prior samples are the inputs of resampling step of PF. The rest steps of our PSO based PF are the same as generic PF.

VI. Simulations

To validate the effectiveness of the proposed tracking approach, some Monte-Carlo simulations have been performed. In these simulations, the target is assumed to move within a square monitored field of size $100m \times 100m$ covered by $N_s = 80$ sensors. The sensing range of each sensor is $r_{sens} = 30m$. Therefore, at each time step, the target is sensed by the sensors which are located at a circle with a radius of $30m$ and the center of the target position. The initial state of the target is $X_0 = [40 \quad -0.5 \quad 40 \quad 0.5 \quad -0.4]^T$ with covariance matrix $P_0 = diag([0.004, 0.002, 0.005, 0.002, 0.0005])$. The BS node is located at the center of the monitored field i.e. (50,50). The covariance matrix of the process noises in NCT model is $Q = diag([0.002, 0.002, 0.002, 0.002, 0.0005])$. The sensors are assumed to have uncorrelated measurement noises with the same variance of $R_i(k) = 0.002$ and the sampling interval is set to $T = 1s$. It should be mentioned that sensors have distance-dependent measurement noise. Therefore the variance of measurement noise of each sensor is multiplied by their relative distance to the target and will be much greater than R_i.The number of particles in PF and PSO is $N_p = 100$ and $N_{pso} = 20$ respectively. The PSO parameters are $\chi = 0.7, c_1 = c_2 = 0.2$. Weighting factors α_1 and α_2 are both considered to be 0.5. Performance of the proposed approach is compared with that of EKF state estimator. For this purpose, we have carried out 100 simulations for both EKF and PSO-PF approaches. Fig. 1 shows true and estimated trajectory of the target using the proposed approach of this paper for one of 100 simulations. It can be seen that the PSO-PF method has estimated the target trajectory accurately. Fig. 2 illustrates how the sensors are deployed in the monitored environment and which sensors are close enough to the target to sense it. Fig. 3 demonstrates state evolutions of the target for both EKF and PSO-PF estimators for one of 100 simulations. It is obvious that PSO-Pf method has a better estimation of target states. To have a more detailed comparison, RMS errors of EKF and PSO-PF estimators have been computed for 100 simulations and given in Table 1.

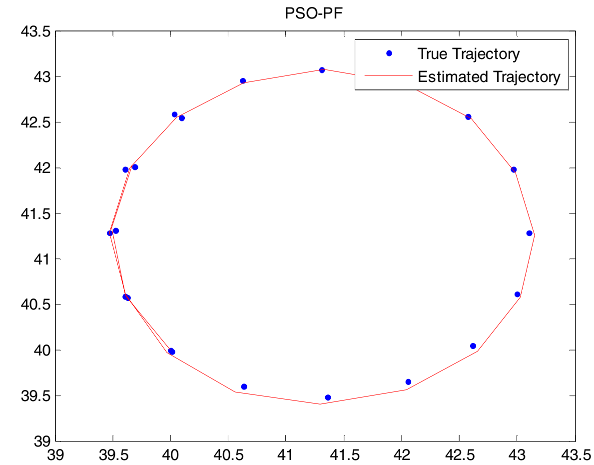

Fig. 1. True and estimated trajectory of the target using PSO-PF estimator

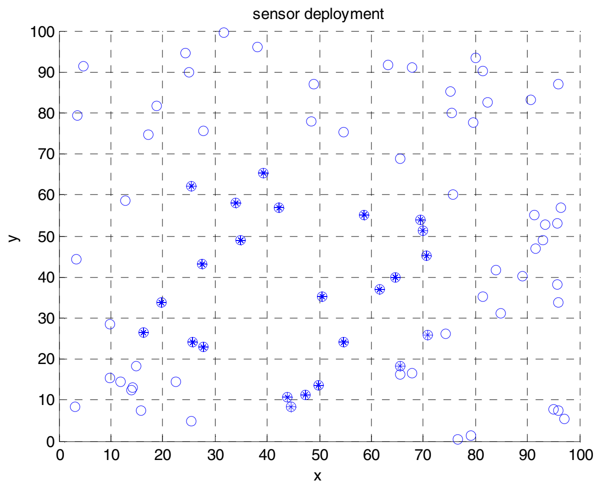

Fig. 2. Sensor deployment (circles with a star in them show the sensors which the target is whitin their range during target tracking)

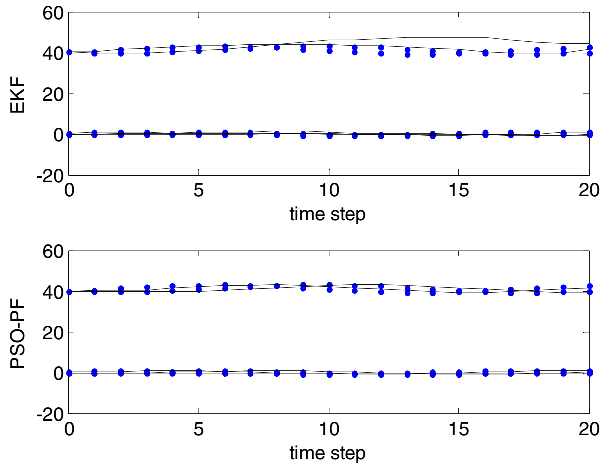

Fig. 3. State estimations of PSO-PF and EKF

TABLE I. RMS ERRORS OF PSO-PF AND EKF ESTIMATORS

PSO-PF RMS error	EKF RMS error
0.5968	4.7934

Table 1 shows that our developed target tracking approach has a lower RMS error than EKF.

VII. CONCLUSION

In this paper, a PSO-PF target tracking approach in range-based WSN with distance-dependent measurement noise was developed. The observation model and state transition model in NCT motion were nonlinear. Therefore, PF was proposed to estimate the target state. Moreover, the distance-dependent measurement noise was incorporated in observation model as a multiplicative noise. The state transition PDF in both NCV and NCT models were derived and the likelihood function was obtained in the case of multiplicative noise. Then, to overcome the impoverishment problem, PSO was merged into PF before the resampling step to move the prior samples towards regions of the state space where both the likelihood and the prior are significant. Simulations results show the effectiveness of the proposed target tracking approach.

REFERENCES

[1] A. Doucet, S. Godsill, and C. Andrieu, “On sequential Monte Carlo sampling methods for Bayesian filtering,” Stat. Comput., vol. 10, no. 3, pp. 197–208, 2000

[2] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking,” IEEE Transactions on Signal Processing, vol. 50, pp. 174–188, 2002.

[3] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, “Particle filters for positioning, navigation, and tracking,” IEEE Transactions on Signal Processing, vol. 50, no. 2, pp. 425–437, 2002.

[4] B. Ristic, S. Arulampalam, and N. Gordon, “Beyond the Kalman Filter, Artech House, 2004.

[5] R. Van der Merwe, A. Doucet, N. De Freitas, and E. Wan. “The unscented particle filter,” Technical report, Cambridge University, Department of Engineering, 2000.

[6] S. J. Julier and J. K. Uhlmann, “A new extension of the kalman filter to nonlinear systems,” In Proceedings of Aerosense: The 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls, vol. Multi Sensor Fusion, Tracking and Resource Management II, Orlando, Florida, 1997.

[7] C.Y. Li, and H.B. Ji, “A new particle filter with GA-MCMC resampling,” IEEE International Conference on Wavelet Analysis and Pattern Recognition, vol. 1, pp. 146-150, 2007.

[8] N. M. Kwok, W. Zhou, G. Dissanayke, and G. Fang, “Evolutionary particle filter: re-sampling from the genetic algorithm perspective,” In Proceedings of the IEEE 2005 Int. Conf. on Intelligent Robots and Systems, pp. 2935–2940, 2005.

[9] G. Tong, Z. Fang, and X. Xu, “A particle swarm optimized particle filter for nonlinear system state estimation,” IEEE Congress on Evolutionary Computation, pp. 438–442, 2006.

[10] Q. Wang, L. Xie, J. Liu, and Z. Xiang, “Enhancing particle swarm optimization based particle filter tracker,” Computational Intelligence, Lecture Notes in Computer Science, vol. 4114, pp. 1216–1221, 2006.

[11] Y. Zheng, and Y. Meng, “Swarming particles with multi-feature model for free-selected object tracking,” IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2553 – 2558, 2008.

[12] X. Zhang, W. Hu, and S. Maybank, “A Smarter Particle Filter,” Computer Vision, Lecture Notes in Computer Science, vol. 5995, pp. 236-246, 2010

[13] A. D. Klamargias, K. E. Parsopoulos, Ph. D. Alevizos, and M. N. Vrahatis, “Particle filtering with particle swarm optimization in systems with multiplicative noise,” In proceedings of the 10th annual conference on Genetic and evolutionary computation, pp. 57-62, 2008.

[14] R. C. Eberhart, and J. Kennedy, “A new optimizer using particle swarm theory,” In Proceedings of Sixth Symposium on Micro Machine and Human Science, pp. 39–43, 1995.

[15] K. E. Parsopoulos, and M. N. Vrahatis, “Recent approaches to global optimization problems through particle swarm optimization,” Natural Computing, vol. 1, pp. 235–306, 2002.