FULL PAPER
International Journal of Recent Trends in Engineering, Vol 2, No. 5, November 2009

Swarm Intelligence Based Tuning of Unscented Kalman Filter for Bearings Only Tracking

Ravi Kumar Jatoth[1], Dr. T. Kishore Kumar[2]
[1, 2] Department of ECE, National Institute of Technology-Warangal, India
Email: [1]ravikumar@nitw.ac.in,[2]kishoret@nitw.ac.in

***Abstract*— Kalman filter is a well known adaptive filtering Algorithm, widely used for target tracking applications. When the system model and measurements are non linear, variation of Kalman filter like extended Kalman filter (EKF) and Unscented Kalman filters (UKF) are used. For obtaining reliable estimate of the target state, filter has to be tuned before the operation (off line).Tuning an UKF is the process of estimation of the noise covariance matrices from process data. In practical applications, due to unavailable measurements of the process noise and high dimensionality of the problem tuning of the filter is left for engineering intuition. In this paper, tuning of the UKF is investigated using Particle Swarm Optimization (PSO) and Bacterial Foraging Optimization (BFO). The simulation results show the superiority of the PSO tuned UKF over the conventional UKF and BFO tuned UKF.**

Index Terms*— Unscented Kalman Filter, *Tracking, Noise* Covariances*, Tuning, Particle Swarm Optimization, BFO.

I. Introduction

In many tracking applications Kalman Filter (KF) is used to estimate the velocity, position and acceleration of a manoeuvring target from noisy radar measurements at high data rates. Bearings only tracking is attracted many researches in these days due to its practical military and civilian applications [1-2]. When the process is to be estimated and measurement model is nonlinear, EKF is used, in which, the process is approximated to first order term of the Taylor's expansion for calculating the mean and covariance of the random process [3].In this process the information present in higher order harmonics can be loosed. Instead of approximating non linear function, if probability distribution function is approximated, then the estimation is more accurate [4]. Approximating a Gaussian distribution is easier than approximating a nonlinear transformation, so state distribution is approximated by a Gaussian random vector.

The Kalman filter demands priori information about the noise covariances from the user [5]. Initial process and measurement noise covariances play an important role in convergence of the filter. If the noise covariances are not chosen properly it may leads towards degradation of the filter performance [6]. A few techniques for determining the process and measurement noise covariances for various applications have been discussed in the literature [7] and widely used tuning method is least squares approach.

This paper investigates the tuning of UKF based on biologically inspired evolutionary computing tools like PSO and BFO.

II. Problem Description

In this paper target tracking environment is taken as shown in figure 1.

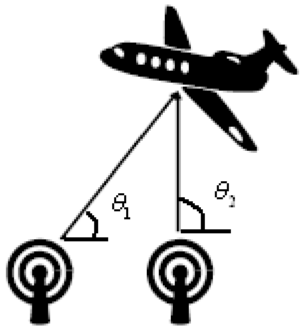

Figure1. Tracking of Plane motion by means of RADAR

We want to track the aircraft position by using sensors. The Problem is to estimate the target kinematic state (position and velocity) from noise corrupted measurements. Since the output of the filtering algorithm is required to be Cartesian position and velocity, the target kinematic state can be described by the state vector defined in discrete time as

$$x_k = [x_k, y_k, v_{xk}, v_{yk}]^T \quad \text{............. (1)}$$

where T denotes matrix transpose, x_k, and y_k are the Cartesian target coordinates at time index k and v_{x_k} and v_{y_k} ,are their respective derivatives (velocities). The state equation for the target motion could be approximated with a linear equation of the form

$$X_{k+1} = F_k x_k + G w_k \quad \text{............ (2)}$$

Where x_k *is the state vector that contains state variables at time k, and* $w_k \sim N(0, Q_k)$ *which is assumed as zero mean white Gaussian noise with covariance* Q_k *(called process noise).*

The state equation for the two dimensional target motion could be approximated with a linear equation of the form

$$\begin{bmatrix} x_{k+1} \\ y_{k+11} \\ v_{x_{k+1}} \\ v_{y_{k+1}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_k & 0 \\ 0 & 1 & 0 & t_k \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ v_{x_k} \\ v_{y_k} \end{bmatrix} + \begin{bmatrix} t_k & 0 \\ 0 & t_k \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_x \\ w_y \end{bmatrix} \quad \text{.......... (3)}$$

Comparing equations (2) and (3), then process noise covariance matrix can be written as

$$Q = E[w_k w_k^T] = \int_0^T G\sigma^2 G^T dt \quad \text{........... (4)}$$

Where σ is the standard deviation of the Gaussian random noise. Whch can be given as

$$= \begin{bmatrix} q_x * t_k^3/3 & 0 & q_x * t_k^2/2 & 0 \\ 0 & q_y * t_k^3/3 & & q_y * t_k^2/2 \\ q_x * t_k^2/2 & 0 & q_x * t_k & 0 \\ 0 & q_y * t_k^2/2 & 0 & q_y * t_k \end{bmatrix}$$

. …………… (5)

Where q_x = level of power spectral density of X-directional noise q_y = level of power spectral density of Y-directional noise.

. The measurement equation above relates the state x_k to the measurement z_k for above scenario sensors are placed at $(s_x^1, s_y^1), (s_x^2, s_y^2)$ and measurement equation can be written as

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} \tan^{-1}(y_k - s_y^1)/(x_k - s_x^1) \\ \tan^{-1}(y_k - s_y^2)/(x_k - s_x^2) \end{bmatrix} + \begin{bmatrix} v_k \\ v_k \end{bmatrix}$$

… ..(6)

Where Vk is White Gaussian noise with the standard deviation Sd.

Measurement noise covariance can be written as

$$R_k = E(v_k v_k^T) \quad ……………… \quad (7)$$

This can be approximated to

$$R_k = dig(sd, sd) \quad ……………… \quad (8)$$

The accuracy of the estimation depends on the priori measurement noise covariance matrix R_k and process noise covariance matrix Q which interns depends upon these spectral densities q_x and q_y. Selecting optimum parameters of these values gives optimum performance of the filter.

Trial and error approach to obtain these the above said three tuning parameters is tedious process and doesn't guarantee the accuracy of estimation in Mean Square Error (MSE) sense. Choosing optimum Parameters of noise covariance matrices, "i.e." is tuning the filter is a challenging task for Kalman filter designer.

In this paper another approach of tuning the Unscented Kalman Filter based on the swarm intelligence is proposed.

III. UNSCENTED KALMAN FILTER

The UKF is a recursive minimum-mean square-error (MMSE) estimator. It is based on the unscented transform (UT). The UT is a method for calculating the statistics of a random variable, which undergoes a nonlinear transformation. State distribution is approximated by Gaussian random vector and is represented by a set of deterministically chosen sample points called sigma points, which completely capture the true mean and covariance of the distribution. High order information about the distribution can be captured using only a very small number of points as problems of statistical convergence are not an issue. Using UT, UKF captures the mean and covariance in the prior and posterior densities accurately [3].

Let L-dimension state vector $\hat{x}_{k-1}$ with mean $\hat{x}_{k-1|k-1}$ and covariance $P_{k-1|k-1}$ be approximated by 2L+1 weighted samples or sigma points. Then one cycle of the UKF is as follows.

Sigma point calculation: Compute the (2L+1) sigma points as follows:

$$\lambda = \alpha^2(L+\kappa) - L \quad (9)$$

$$X^0_{k-1|k-1} = \hat{x}_{k-1|k-1}$$

$$W_0^m = \lambda/(L+\lambda),$$

$$X^i_{k-1|k-1} = \hat{x}_{k-1|k-1} + (\sqrt{(L+\lambda)P_{k-1|k-1}})_i,$$

$$W_i^m = 1/2(L+\lambda), i = 1,...,L,$$

$$X^{i+L}_{k-1|k-1} = \hat{x}_{k-1|k-1} - (\sqrt{(L+\lambda)P_{k-1|k-1}})i,$$

$$W_{i+L}^m = 1/2(L+\lambda), i = 1,...L,$$

$$W_0^c = W_0^m + (1 - \alpha^2 + \beta)$$

$$W_i^m = W_i^c, i = 1,....,2L$$

Where α determines the spread of sigma points around the mean and is usually set to a small positive value, κ is a secondary scaling parameter which is usually set to 3-L, β is used to incorporate prior knowledge of distribution of x and $\left(\sqrt{(L+\lambda)P_{k-1|k-1}}\right)_i$ is the i[th] row or column (depending on the matrix square root form, if $P = A^T A$ then the sigma points are formed from the rows of A. However, if the matrix square root is of the form $P = AA^T$, the columns of A are used) of the matrix square root of $(L+\lambda)P_{k-1|k-1}$ and W_i is the normalized weight associated with the i[th] point. Note that Cholesky decomposition is needed for the matrix square root.

Propagation: Propagate the sigma points and obtain the mean and covariance of the state by

$$X^i_{k|k-1} = f(X^i_{k-1|k-1}) \quad (10)$$

$$\hat{x}_{k|k-1} = \sum_{i=0}^{2L} W_i^m X^i_{k|k-1} \quad (11)$$

$$P_{k|k-1} = Q_{k-1} + \sum_{i=0}^{2L} W_i^c \left[X^i_{k|k-1} - \hat{x}_{k|k-1}\right] \times \left[X^i_{k|k-1} - \hat{x}_{k|k-1}\right]^T \quad ..(12)$$

Update: Calculate the measurement sigma points $Z^i_{k|k-1}$ using h(.) and update the mean and Covariance by

$$Z^i_{k|k-1} = h(X^i_{k|k-1}) \quad ………. (13)$$

$$\hat{z}_{k|k-1} = \sum_{i=0}^{2L} W_i^m Z^i_{k|k-1} \quad ….. (14)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k[z_k - \hat{z}_{k|k-1}] \quad ……(15)$$

$$P_{k|k} = P_{k|k-1} - K_k P_{zz} K_k^T, …… \quad (16)$$

$$P_{zz} = R_k + \sum_{i=0}^{2L} W_i^c \left[Z^i_{k|k-1} - \hat{z}_{k|k-1}\right] \times \left[Z^i_{k|k-1} - \hat{z}_{k|k-1}\right]^T .. \quad (17)$$

$$P_{xz} = \sum_{i=0}^{2L} W_i^c \left[X^i_{k|k-1} - \hat{x}_{k|k-1}\right] \times \left[Z^i_{k|k-1} - \hat{z}_{k|k-1}\right]^T ..(18)$$

$$K_k = P_{xz}P_{zz}^{-1} \quad …….. \quad (19)$$

The Jacobian matrices are not required to implement this algorithm. The other advantage of the UKF over the EKF is that it can estimate the mean and covariance of the state accurately to second order for any nonlinearity.

IV. PARTICLE SWARM OPTIMIZATION

PSO is population based stochastic optimization technique inspired by social behavior of bird flocking or fish schooling etc[8]. The swarm of particles indicates estimation of multiple parameters involved in the problem. We can begin with initializing a random swarm of particles. During each iteration fitness of the particle is evaluated with the help of fitness function.

The trajectory of the particle is dependent on three factors: its previous position, pbest and gbest. Greater the strain of particle in searching food, smaller is the acceleration coefficients. The inertial weight factor w signifies the importance of the particle's previous position in further search.

Velocity updation

$$v_i(t+1) = w.v_i(t) + c_1 rand(pbest(t) - x(t)) + c_2.rand.(gbest(t) - x_i(t)) \quad (20)$$

Position updation

P=P+V …………… (21).

Where

P - Instantaneous position of the particle
V - Instantaneous velocity of the particle
Pbest - positional best of the particle
gbest – global best position of the swarm of the particles
W – Inertial weight factor
C1, C2 – acceleration coefficients

Thus each particle tends to move towards gbest to reach food early. If gbest has less number of values then the particles will reach food early. The algorithm comes to an end when all the particles converge at the gbest i.e. food position [8]. In our problem i.e. attaining minimum possible value for steady state error signal is considered as global optimum. The flow chart for PSO can be shown in figure 2.

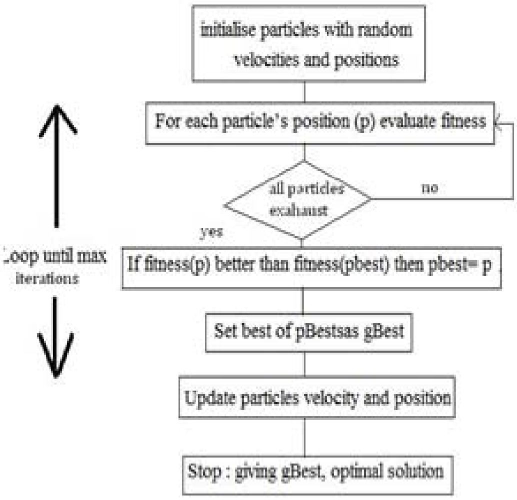

Figure 2: Flow chart for PSO Algorithm

V. FILTER TUNING

A. Importance of Filter Tuning

Tuning of the filter is referred as as a process of obtaining parameters of a filter such as values of matrices *Q* and *R* for EKF that give the best filter performance in Mean Square Error (MSE) sense [9]. Typically this kind of problems of designing a filter with optimal tuning parameters was left up to engineering intuition, and trial and error method that do not guarantee best filter performance due to large number of parameters to be tuned. A straightforward way of tackling this problem is to employ global optimization method that minimizes function of MSE position error with respect to filter parameters. There are several issues associated with such an approach. First, each time we need a value of MSE during global optimization procedure we have to run UKF on all available data. This requires a significant computational time since for example in order to find a global minimum of a smooth function of 3 parameters; we need to compute the function value many times.

B. *Applying PSO Filter Tuning*

One of the practical solutions to these issues is to estimate approximate functional relation between tuning parameters and the MSE criterion of optimization in a deterministic way and then apply nonlinear global optimization method to find optimal parameters which correspond to minimum of MSE.

Here in this problem we are tracking with constant velocity and with small maneuver such as to relate practical problem. Therefore we have two power spectral densities of the corresponding continuous process noise, one parameter of measurement noise standard deviations (bearing). So a total of three parameters have to be optimized. Taking the extreme worst cases of these three parameters, we proceed according to the Particle Swarm Optimization.

Initialization of PSO

Size of the swarm " no of birds=30;
Maximum number of "birds steps=30;
Dimension of the problem =3;
PSO parameter C1= 2.05

PSO parameter C2 =2.05
pso momentum or inertia w= 0.45
search space for Sd= 0 t0 0.1
search space for q_x= 0 t0 0.01
search space for q_x= 0 t0 0.05

VI. BACTERIAL FORAGING OPTIMIZATION

The Bacterial Foraging optimization is based upon search and optimal foraging decision making capabilities of the E.Coli bacteria. Each position of the bacterium represents an individual solution of the optimization problem. Such a set of trial solutions converges towards an optimal solution following the foraging group dynamics of the bacterial population.

The bacterial foraging system primarily consists of four sequential mechanisms namely chemo taxis, swarming, reproduction and elimination-dispersal. A brief outline of each of these processes is given in this section.

(1) **Chemo taxis:** An E. coli bacterium can move in two different ways: it can run (swim for a period of time) or it can tumble, and alternate between these two modes of operation in the entire lifetime. In the BFO, a unit walk with random direction represents a tumble and a unit walk in the same direction indicates a run. In computational chemotaxis, the movement of the ith bacterium after one step is represented as

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\phi(j)$$

Where $\theta(j,k,l)$ denotes denotes the location of i^{th} bacterium at jth chemotactic, kth reproductive and lth elimination and dispersal step. C(i) is the length of unit walk, and $\phi(j)$ is the direction angle of the jth step.

(2) **Swarming:** The bacteria in times of stresses release attractants to signal bacteria to swarm together. Each bacterium also releases a repellant to signal others to be at a minimum distance from it. Thus all of them will have a cell to cell attraction via attractant and cell to cell repulsion via repellant. The cell to cell signaling in E. coli swarm may be mathematically represented as

$$J_{CC}(\theta, P(j,k,l)) = \sum_{i=1}^{s} Jcc(\theta, \theta^i(j,k,l))$$
$$= \sum_{i=1}^{s} -d_a \exp(-w_a \sum_{m=1}^{p} (\theta_m - \theta_m^i)^2)$$
$$+ \sum_{i=1}^{s} -h_r \exp(-w_r \sum_{m=1}^{p} (\theta_m - \theta_m^i)^2)$$

where Jcc(θ, P(j,k, l)) represents the objective function value to be added to the actual objective function, S is the total number of bacteria, p is the number of variables to be optimized and $\theta = [\theta_1, \theta_2, . . ., \theta p]^T$ is a point in the p-dimensional search domain. da, wa, hr and wr are coefficients to be chosen properly.

(**3) Reproduction:** After all Nc chemotactic steps have been covered, a reproduction step takes place. The fitness values of the bacteria are sorted in ascending order. The lower half of the bacteria having higher fitness die and the remaining Sr = S/2 bacteria are allowed to spilt into two identical ones. Thus the population size after reproduction is maintained constant.

(4) Elimination and dispersal: Since bacteria may stuck around the initial or local optima positions, it is required to diversify the bacteria either gradually or suddenly so that the possibility of being trapped into local minima is eliminated. The dispersion operation takes place after a certain number of reproduction process. A bacterium is chosen, according to a preset probability Ped, to be dispersed and moved to another position within the environment. These events may help to prevent the local minima trapping effectively, but unexpectedly disturb the optimization process.

Initialization of BFO

Dimension of search space=3;
The number of bacteria =10;
Number of chemotactic steps =6;
The number of reproduction steps =3
The number of elimination-dispersal events=3
The number of bacteria reproductions (splits) per generation=3
The probabilty that each bacteria will be eliminated/dispersed =0.2

VII. SIMULATIONS AND RESULTS

Here we consider a target scenario in which a moving target in the scene and two angular sensors for tracking it. The sensors are placed to (s_x^1, s_y^1) = (-1m,-2m) and (s_x^2, s_y^2) = (1m, 1m). The measurement noise standard deviation is taken as Sd= 0.5 radians and spectral densities of the process noise is consider as q_x=0.1 and q_y =0.1 to generate data as show in figure 4 below. The simulations are performed using industry standard MATLAB and EKF/UKF Toolbox.

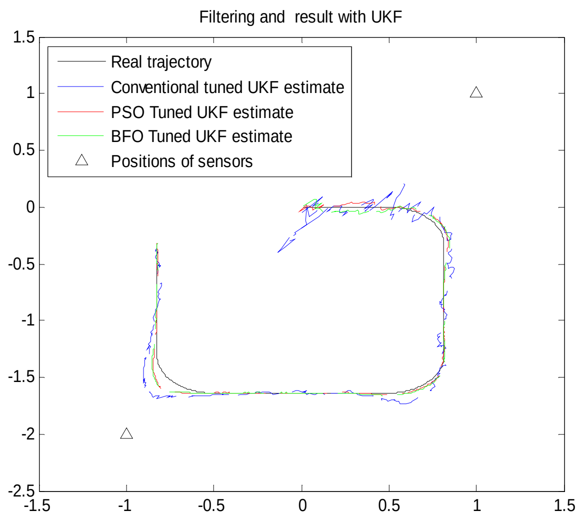

Figure3: Filtering results of conventional and PSO-Tuned UKF

From figure 3 we can see that the PSO Tuned UKF is tracking the object accurately compared to coventional UKF and BFO tuned UKF.

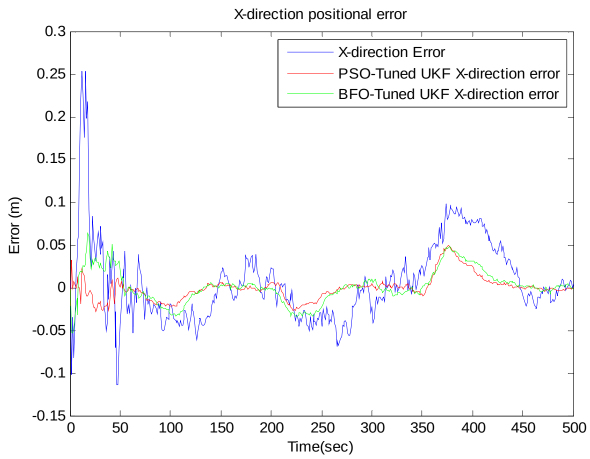

Figure 4: Positional error in X-direction.

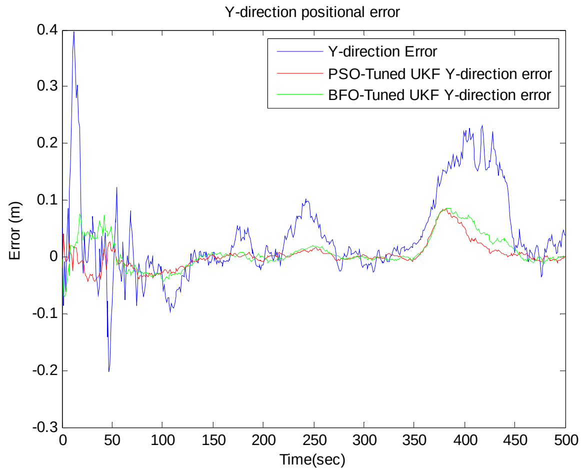

Figure 5: Positional error in Y-direction

From the above figures we can say that error in both the direction X and Y is reduced using PSO-Tuned UKF.

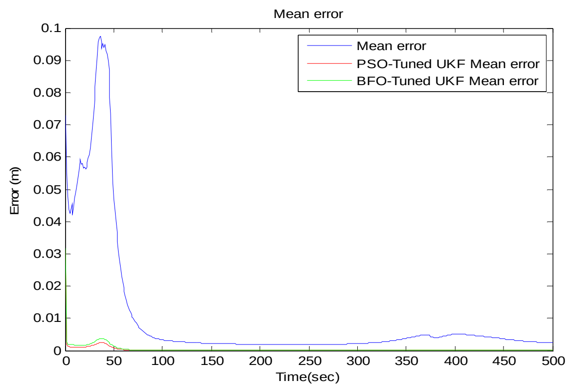

Figure 6: Mean error

From the figure 6 shows the quality of the estimations as the no of measurements precedes the filter is converging.

IX .CONCLUSIONS AND FUTURE SCOPE

The work presents tuning Procedure for UKF. A comparison was made between two nonlinear filtering algorithm standard UKF, PSO and BFO Tuned UKF for maneuvering target tracking. Since the measurement covariance can be determined in different environments, like off-line, we can get standard deviation for different conditions. Then, PSO-Tuned UKF can be applied for fine tuning of noise covariance matrices. The results are shown for conventional tuned, BFO and PSO-tuned UKF.

Parameter	Conventional UKF	BFO-Tuned UKF	PSO-Tuned UKF
Measurement noise standard deviation(Sd)	0.05	0.0052	0.0050
q_x	0.1	0.0120	0.0129
q_y	0.1	0.0115	0.0136
RMSE (m)	0.099	0.035	0.027
Computational complex city	0.00156 (Seconds)	0.622 (Seconds)	0.424 (Seconds)

We have presented a new approach for estimating state in nonlinear system. From the results we can conclude that PSO tuned UKF gives better tuning of UKF. Given its performance and implementation advantages in the example, we can conclude that the new filter is promising over conventional tuned and BFO tuned UKF and be efficient in many nonlinear filtering applications. Future work can be focused on any other new evolutionary computing algorithm and can be implemented on DSP or FPGA processor.

References

[1] S.Sadhu,M.Srinivasan and T.Ghosal, "Robustness and Tuning of Bearing only tracking", pp 42-49,IE(I) Signal processing ,Journal,vol 87,Dec-2006

[2] T L Song and J L Speyer "A Stochastic Analysis of Modified Gain Extended kalman filter With Applications to Estimation with Bearing s Only Measurements, IEEE Transactions on Automatic Control, , pp 940-949 vol 30,no 10,1985

[3] Simon J. Julier , Jerey K. Uhlmann "New Extension of the Kalman Filter to Nonlinear Systems" *The Robotics Research Group, Department of Engineering Science, The University of Oxford*, Oxford, OX1 3PJ, UK

[4] Simon J. Julier, Jeffery K. Uhlmann, "Unscented filtering and nonlinear estimation," Proc. IEEE 92 (3) March 2004.

[5] Bernt M. A ˚ kesson, John Bagterp Jørgensen, Niels Kjølstad Poulsen, A generalized auto covariance least-squares method for Kalman filter tuning, *A journal of process control , Elsevier 2007.*

[6] A Tool for Kalman filter Tuning, B.M Akesson 17th European symposium on computer aided process Engineering ESCAPE1.

[7] B.J. Odelson, A. Lutz, J.B. Rawlings, The autocovariance leastsquares method for estimating covariances: applications to modelbased control of chemical reactors, *IEEE Transactions on Control Systems Technology* 14 (3) (2006) 532–540.

[8] Eberhart R C., Kennedy J., A new optimizer using particle swarm theory [A], Proc. *The Sixth Int. Symposium on Micro Machine and Human Science* [C], Nagoya, Japan, 1995, pp. 39-43.

[9] Oleksiy V.Korniyenko, Mohammad S.Sharawi, Oklahand University ,"Neural Network based approach for tuning kalman filter", *2005 IEEE International Conference on Volume* , Issue , 22-25 May 2005 Page(s): 1 - 5.