IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 5, Ver. IV (Sep. - Oct. 2016), PP 14-19
www.iosrjournals.org

Tuning of Extended Kalman Filter for nonlinear State Estimation

Navreet Kaur[1], Amanpreet Kaur[2]
[1, 2](Information Technology Department, Chandigarh Engineering college, Landran, India)

Abstract: *Kalman Filter is the most popular method for state estimation when the system is linear. State estimation is the typical issue in every part of engineering and science. But, for non linear systems, different extensions of Kalman Filter are used. Extended Kalman Filter is famous to discard the non linearity which uses First order Taylor series expansion. But for these estimation techniques, the tuning of process noise covariance and measurement noise covariance matrices is required. There are different optimization techniques used to tune the parameters of Extended Kalman Filter. In this paper, Particle Swarm Optimization has been proposed to tune the EKF parameters and then the simulations are implemented for permanent magnet synchronous motor.*

Keywords: *Extended Kalman Filter, Particle Swarm Optimization, non-linear, state estimation, tuning of EKF.*

I. Introduction

For a Dynamic system, it is mandatory to estimate the state using a sequence of noisy measurements. Somewhat there is need of two models to analyze and make presumption about a dynamic system: First is, a system model, which express the evolution of the state with time and the second is the measurement model, which is related to the noisy measurements of the state [1]. There are various filters used to solve the issue of state estimation. These filters include the Kalman filter, Particle filter, Unscented Kalman Filter, Extended Kalman filter and also a Fokker-Planck equation based constant rate filter [2].

Kalman Filter is used only when the system is linear, but almost all practical systems have non linearity. If the dynamic model is non linear, then the Extended Kalman filter is used to eliminate the non linearity [3]. Extended Kalman Filter is the extended form of Kalman Filter, which is based on linearization of first order Taylor series expansion. By using Gaussian random variable the state distribution has been approximated [4]. The other derivative of Kalman Filter, i.e. Unscented Kalman Filter (UKF) is also used when the system is non linear and it also gives better results than Extended Kalman Filter but it suffers from the problem of divergence. So, Extended Kalman filter is the famous state estimation technique because of its simplicity [5]. Extended Kalman Filter gives an approximation of the optimal estimate. But, it suffers from different problems like divergence, initialization and linearization error and covariance estimation [6]. Therefore, it is necessary to tune the filter parameters of the Extended Kalman Filter, i.e. process noise covariance (Q) and measurement noise (R) covariance matrices.

Almost all the filters require the tuning process after the implementation. In Filter tuning process, the measurement noise and dynamic noise statistics are selected to get better performance [2]. There are different methods used to tune the measurement noise covariance (R) matrix and process noise covariance (Q) matrix. Earlier, an adaptive Kalman filtering approach is used to tune the measurement noise covariance matrix and process noise covariance matrix. Three different schemes of adaptation have been used by the Adaptive Kalman Filter. These include: measurement noise covariance matrix (R), process noise covariance matrix (Q) and the initial values of error covariance matrix (P) [7]. There are two adaptive methods used, i.e. multiple model adaptive estimation in which multiple Kalman filters runs parallel, and innovative adaptive estimation in which the Q and R matrices are adapted by themselves [8]. But these methods have problem of lack of convergence, and large window requirement.

There are different optimization techniques used to tune the Q and R matrices of Extended Kalman Filter. In this paper Particle Swarm Optimization technique is used to tune the Extended Kalman Filter for non linear state estimation. Particle Swarm Optimization is a metamorphic technique which is influenced by the flocks of birds or schools of fishes [10]. Using fitness function the fitness of the particle is calculated.

In Section 2, the Extended Kalman Filter is presented. Section 3 presents the particle swarm optimization Technique and section 4 describes the problem formulation and system design. Section 5 describes the methodology used. Section 6 shows the simulation results. At the end, conclusion is there.

II. Extended Kalman Filter

By considering the following equation, the non linear system model can be described as given in [9],

$$\begin{aligned} ҳ_t &= f(ҳ_{t-1}) + \dot{w}_{t-1} \\ \check{z}_t &= h(ҳ_t) + \dot{v}_t \end{aligned} \tag{1}$$

$ҳ_0$ denotes initial state which is a random vector having mean $\dot{\mu}_0 = E[ҳ_0]$ and covariance $\dot{P}_0 = E[(ҳ_0 - \dot{\mu}_0)(ҳ_0 - \dot{\mu}_0)^T]$. v_t represents the measurement noise and the random vector $\dot{w}_t$ taking the ambiguities in the model. These both are uncorrelated with the initial state $ҳ_0$.

$$\begin{aligned} &E[\dot{w}_t] = 0,\ E\left[\dot{w}_t {\dot{w}_t}^T\right] = Q_T,\ E\left[\dot{w}_t {\dot{w}_j}^T\right] = 0 \text{ for } t \neq j,,\ E[\dot{w}_t {ҳ_0}^T] = 0 \text{ for all } t \\ &E[\dot{v}_t] = 0,\ E\left[\dot{v}_t {\dot{v}_t}^T\right] = R_T,\ E\left[\dot{v}_t {\dot{v}_j}^T\right] = 0 \text{ for } t \neq j,,\ E[\dot{v}_t {ҳ_0}^T] = 0 \text{ for all } t \end{aligned} \tag{2}$$

$\dot{v}_t$ and $\dot{w}_t$ both are also uncorrelated,

$$E\left[\dot{w}_t {\dot{v}_j}^T\right] = 0 \text{ for all } t \text{ and } j$$

The victorial functions $f(.)$ and $h(.)$ are pretended to be C^1 functions i.e. the functions and its first derivatives are continuous on the given domain.

Following are the dimension and description of the variables:

$ҳ_t$ $n \times 1$ is the state vector, $\dot{w}_t$ $n \times 1$ is the process noise vector, $\check{z}_t$ $m \times 1$ is the observation vector, $\dot{v}_t$ $m \times 1$ is the measurement noise vector, $f(.)$ $n \times 1$ is the process non linear vector function, $h(.)$ $m \times 1$ observation non linear vector function, Q_t $n \times n$ is the process noise covariance matrix, R_t $m \times m$ is the measurement noise covariance matrix.

By expanding the $f(ҳ_t)$ and $h(ҳ_t)$ using Taylor series, the forecasting and prediction steps are approximated.

Initialization step:

$ҳ_0^a = \dot{\mu}_0$ with error covariance $\dot{P}_0$

Data Forecasting Step:

$$\begin{aligned} ҳ_t^f &\approx f(ҳ_{t-1}^a) \\ \dot{P}_t^f &= J_f(ҳ_{t-1}^a)\dot{P}_{t-1} J_t^T(ҳ_{t-1}^a) + Q_{t-1} \end{aligned} \tag{3}$$

Data Assimilation Step:

$$\begin{aligned} ҳ_t^a &\approx ҳ_t^f + K_t\left(z_t - h\left(ҳ_t^f\right)\right) \\ K_t &= \dot{P}_t^f J_h^T\left(ҳ_t^f\right)\left(J_h\left(ҳ_t^f\right)\dot{P}_t^f J_h^T\left(ҳ_t^f\right) + R_t\right)^{-1} \\ \dot{P}_t &= \left(I - K_t J_h\left(ҳ_t^f\right)\right) + \dot{P}_t^f \end{aligned} \tag{4}$$

Where, K_t is the Kalman Gain, J_h is the jacobian of $h(.)$ and J_f is the jacobian of $f(.)$.

III. Particle Swarm Optimization

It is a metamorphic optimization technique which is inspired by mutual nature of flocking of birds and schooling of fishes. In PSO, "particle" is somewhere in the search space and fitness value of all the particles should be calculated using the fitness function, and also velocity should be calculated, which direct the flying of the particles. By achieving the current optimum particles, the particles fly through the problem space [10].

There are always two states in which the particle is illustrated and that are denoted as y_i as position and $ύ_i$ as velocity of the i^{th} particle. In each iteration, the states of every i^{th} particle are renewed using following equation,

$$\begin{aligned} ύ_i(ŧ + dŧ) &= ẅ \times ύ_i(ŧ) + c_1 \times r_1 \times \left(pbest_i(ŧ) - y_i(ŧ)\right) + c_2 \times r_2\left(gbest(ŧ) - y_i(ŧ)\right) \\ y_i(ŧ + dŧ) &= y_i(ŧ) + ύ_i(ŧ)dŧ \end{aligned} \tag{5}$$

where, $pbest_i$ indicates the best position which is attained by i^{th} till present, $gbest$ shows the best position which is attained by any particle untill present, c_1 and c_2 are acceleration vectors, r_1 and r_2 are random vectors consistently distributed between '0' and '1', ẅ represents the inertial weight.

IV. Problem Formulation and Experimental Design

For the indirect measurement of rotor's speed and pole position, the permanent magnet synchronous motor is used [11]. To accurately evaluate the rotor speed and rotor position, EKF is implemented. The problems related to the estimation of rotor speed and positions via electromechanical devices are overcome using indirect measurement through voltages and currents of the phases. Permanent magnet is used to mount the rotor. The given equations defined the process model dynamics:

$$\begin{gathered}
\dot{\text{ɕ}}_a = \frac{-\text{Я}}{I}\text{ɕ}_a + \frac{\text{ʂ}\gamma}{I}\sin\theta + \frac{(v_a + n_1)}{I} \\
\dot{\text{ɕ}}_b = \frac{-\text{Я}}{I}\text{ɕ}_b - \frac{\text{ʂ}\gamma}{I}\cos\theta + \frac{(v_b + n_2)}{I} \\
\text{ʂ}_a = \frac{-3\gamma}{2\mathcal{M}}\text{ɕ}_a \sin\theta + \frac{3\gamma}{2\mathcal{M}}\text{ɕ}_b \cos\theta - \frac{B\text{ʂ}}{\mathcal{M}} + n_3 \\
\theta \doteq \text{ʂ}
\end{gathered} \tag{6}$$

here, ɕ_a and ɕ_b are two phase currents, ʂ is rotor speed and θ is the position. Я is the per phase resistance and I is the per phase inductance of the winding. γ is the flux constant, , v_a and $v_b$denotes applied voltages across the winding, B is the viscous damping. $\mathcal{M}$ is the moment of inertia of the rotor. n_1 and n_2 are noise in the control voltages, $n_3$is a load torque. For AWGN process, all noises are assumed to be zero mean band-limited.

Extended Kalman Filter Model is simulated on MATLAB R2013 for input control voltages which is given by $v_a = \sin \text{ʂ}t$and $v_b = \cos \text{ʂ}t$, for which the output current ɕ_a and ɕ_b are calculated. $u = [\text{ɕ}_a \text{ɕ}_b \text{ʂ}\ \theta]T$ represents the state vectors. $z = [\text{ɕ}_a \text{ɕ}_b]T$ represents the measurement matrix, $\dot{r} = f(r)$ is the process model and, $z = \text{Ⱨ}_r$ is the measurement model which are utilized. Ⱨ is defined as [1 0 0 0; 0 1 0 0]. By executing Jacobian operator on $f(r)$, F is computed by given equation

$$F = \frac{\partial f}{\partial r}\begin{bmatrix} {}^{-\text{Я}}/_{I} & 0 & {}^{\lambda}/_{I} & {}^{\lambda\text{ʂ}}/_{I} \\ 0 & {}^{-\text{Я}}/_{I} & {}^{-\lambda}/_{I} & {}^{\lambda\text{ʂ}}/_{L} \\ {}^{-3\lambda}/_{2J} & {}^{3\lambda}/_{2J} & {}^{-B}/_{J} & {}^{-3\lambda(\text{ɕ}_a + \text{ɕ}_b)}/_{2J} \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{7}$$

For very noisy environment, system is simulated and in phase current, the standard deviation of the control noise (σ_{c0}) is taken as 10, load torque noise (σ_{T0}) is 0.5 and measurement noise (σ_{m0}) taken as 500 for system simulation. The values of Q and R are adjusted by varying the standard deviation of the measurement noise and process which are given as

$$Q = \begin{bmatrix} \left({}^{\sigma_c}/_{I}\right)^2 & 0 & 0 & 0 \\ 0 & \left({}^{\sigma_c}/_{I}\right)^2 & 0 & 0 \\ 0 & 0 & {\sigma_\tau}^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, R = \begin{bmatrix} {\sigma_m}^2 & 0 \\ 0 & {\sigma_m}^2 \end{bmatrix} \tag{8}$$

To improve the filter performance, there is a need to tune the Q and R filter parameters. Therefore, the evolutionary techniques for optimization are used to tune the filter parameters. Thus, Particle Swarm Optimization technique is used to tune the filter parameters of the Extended Kalman filter.

V. Methodology

The Particle Swarm Optimization is proposed here to tune the filter parameters of the Extended Kalman Filter to improve the performance. For the values σ_c and σ_m, an opinion vector is formed and to minimize the given cost function, the filter is run

$$J_c = \sqrt{\rho^2{}_{error} + \theta^2{}_{error}} \tag{9}$$

The above function gives a fitness value which is equal to the value of the cost function.
The pseudo code for PSO based tuning of EKF is shown below:

```
I.    Initialize each particle $u_i$ for every dimension by distributing them randomly between ($U_{min}$, $U_{max}$).
II.   FOR it= 1 to 100.
III.  FOR i = 1 to 100.
IV.   Initializing the filter by appoint values Q and R which are find out by particles and then the EKF is run.
V.    Determine the error value which corresponds to $i^{th}$ particle
VI.   IF error identifies at $i^{th}$ iteration < pbest at $(i-1)^{th}$ iteration
          Update p-best by the particle
          Store the fitness value as p-best_fitness
          ELSE
          Update p-best by previous p-best.
VII.  END
VIII. Update g-best by the comparable p-best having minimum of p-best_fitness
IX.   Update particle velocity and position.
X.    IF maximum iteration condition or minimum error condition is attained
XI.   Break from loop
XII.  END
```

XIII. Simulation Results

By using hit and trail method, an approximate values of σ_c and σ_m are obtained so that the Kalman filter does not depend on the process and measurement model which results zero or infinite value of Kalman gain. A plot of the true and estimated values is shown in the Fig.1 for $\sigma_{c0} = \sigma_c$ and $\sigma_{m0} = \sigma_m$. In the figure, it is shown that the estimated state does not match with the true state of the system.

For PSO based Extended Kalman Filter tuning, the value of σ_c is uniformly distributed from 0.1 to 10 and the value of σ_m is distributed from 500 to 10000. In this simulation, the number of particle vectors taken are 100 and each particle should be allow to move in the search space and then updated in every iteration. The values of vector $[\varsigma_1\ \varsigma_2]$ is taken as [0.2 0.02] and the value of scalar n is 0.9.

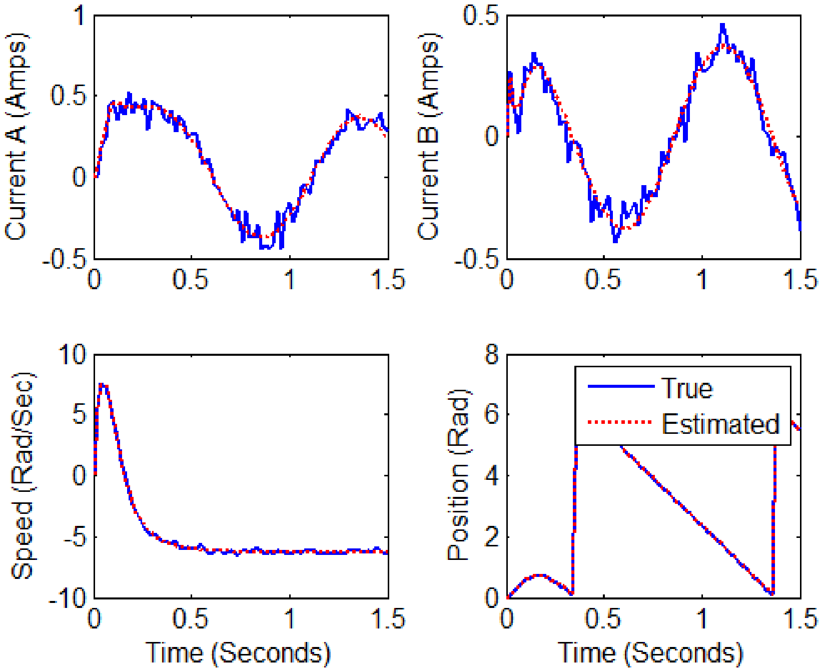

Fig.1 Results for PSO based EKF tuning for these states; Current (ς_a), Current (ς_b),Speed, Position

The values found by using PSO based EKF tuning are: $\sigma_c = 4.6574$, $\sigma_m = 6140.6$ and e =0.0522. The mean square error value of particles is shown in figure 3 which is plotted for 100 iterations.

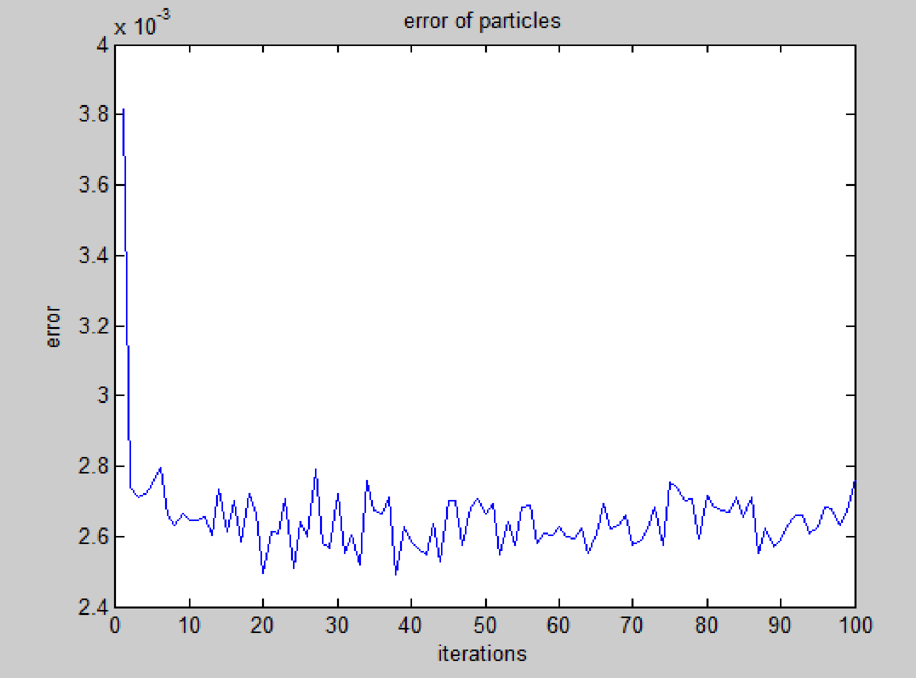

Fig.2 Error value for particles for PSO based EKF

The standard deviation value of particle position is shown in figure 3 which is nearly 0.005.

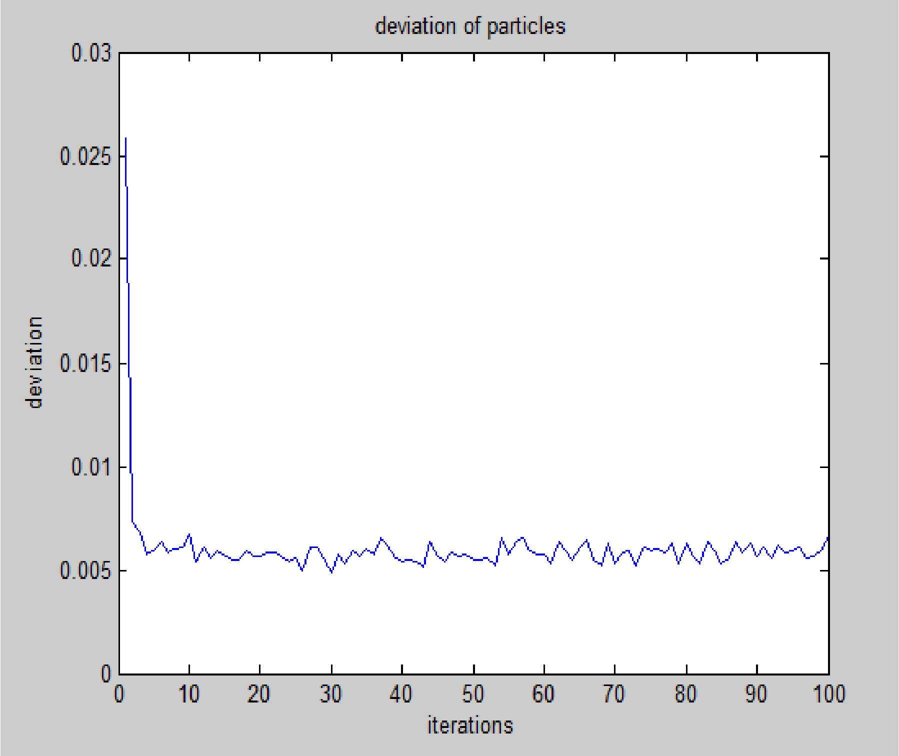

Fig.3 Standard deviation value for particles in PSO

Figure 4 shows the convergence of particles which have their own velocity and which converge towards the global best position for food. From the figure 4, it is seen that the number of iterations taken by the particles to get converge are approximately 100.

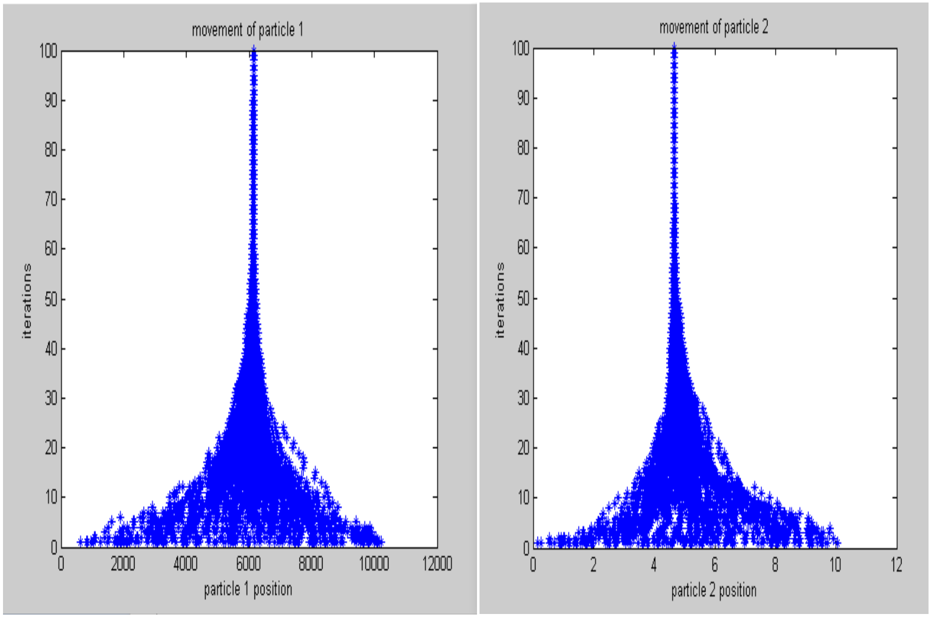

Fig.4 convergence of particles

IX. Conclusion

In this paper tuning of Extended Kalman Filter parameters has been done using Particle Swarm Optimization. Particle Swarm Optimization provides better performance using tuning process model. Simple Extended Kalman Filter does not provide the satisfying results but after tuning it shows better results.

Acknowledgements

I am thankful to my guide Mrs. Amanpreet Kaur for her guidance to complete this research paper.

References

[1]. F. Abdallah, A. Gning and P. Bonnifait, Box particle filtering for nonlinear state estimation using interval analysis, *Automatica*, *44*(3), 2008, 807-815.

[2]. DM. Asmar and Greg J. Eslinger, Nonlinear programming approach to filter tuning, 2012.

[3]. K. György, A. Kelemen and L. Dávid, Unscented kalman filters and particle filter methods for nonlinear state estimation, *Procedia Technology*,*12*, 2014, 65-74.

[4]. 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*, *50*(2), 2002, 174-188.

[5]. K. Xiong, H. Zhang and L. Liu, Adaptive robust extended Kalman filter for nonlinear stochastic systems, *IET Control Theory & Applications*,*2*(3), 2008, 239-250.

[6]. D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*, John Wiley & Sons, 2006.

[7]. A. Almagbile, J. Wang and W. Ding, Evaluating the performances of adaptive Kalman filter methods in GPS/INS integration, *Journal of Global Positioning Systems*, *9*(1), 2010, 33-40.

[8]. D. Loebis, R. Sutton, J. Chudley, and W. Naeem, Adaptive tuning of a Kalman filter via fuzzy logic for an intelligent AUV navigation system, *Control engineering practice,* 12(12), 2004, 1531-1539.

[9]. G. A. Terejanu, Extended kalman filter tutorial. *Online]. Disponible: http://users. ices. utexas. edu/~ terejanu/files/tutorialEKF. Pdf*, 2008.

[10]. J. Zhao and Z. Li, Particle filter based on Particle Swarm Optimization resampling for vision tracking. *Expert Systems with Applications 37*(12), 2010, 8910-8914.

[11]. R. Dhaouadi, N. Mohan and L. Norum, Design and implementation of an extended Kalman filter for the state estimation of a permanent magnet synchronous motor. *Power Electronics, IEEE Transactions on 6*(3), 1991, 491-497.