Defence Science Journal, Vol. 71, No. 4, July 2021, pp. 556-563, DOI : 10.14429/dsj.71.16284

Application of Sigma Point Particle Filter Method for Passive State Estimation in Underwater

G. Naga Divya[*] and S. Koteswara Rao

Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur - 522 502, India
[*]E-mail: nagadivya.guduru@gmail.com

ABSTRACT

Bearings-only tracking (BOT) plays a vital role in underwater surveillance. In BOT, measurement is tangentially related to state of the system. This measurement is also corrupted with noise due to turbulent underwater environment. Hence state estimation process using BOT becomes nonlinear. This necessitates the use of nonlinear filtering algorithms in place of traditional linear filters like Kalman filter. In general, these nonlinear filters utilize the assumption of measurements being corrupted with Gaussian noise for state estimation. The measurements cannot be always corrupted with Gaussian noise because of the highly unstable sea environment. These problems indicate the necessity for development of nonlinear non-Gaussian filters like particle filter (PF) for underwater tracking. However, PF suffers from severe problems like sample degeneracy and impoverishment and also it is tedious to select an appropriate technique for resampling. To overcome these difficulties in PF implementation, the strategy of combining PF with another filter like unscented Kalman filter is proposed for target's state estimation. The detailed analysis of the same is presented in comparison with other particle filter combinations using the simulation results obtained in Matlab.

Keywords: Bearings-only tracking; Unscented Kalman filter; Particle filter; State estimation

1. INTRODUCTION

Nonstationary inverse problems commonly referred as state estimation problems needs a lot of attention due to its numerous real time applications such as air and underwater surveillance, computer vision and target tracking etc. In these problems, the state dynamic variables of interest are estimated sequentially with the use of obtainable measured data along with previous knowledge of the physical phenomenon. It is performed in such a manner that the error/variance is significantly reduced.

Surveillance in underwater is commonly carried out by Bearings-only tracking[1] (BOT). In underwater BOT, the observer and target can be a ship or submarine. The measurement, bearing is an angle measured in the direction of the clock, between the line joining observer and target with true north. It is generated by passive sonar arranged on an observer's hull using the noise produced by target. The bearing angle is tangential (nonlinearly) related with the state vector. The measurements are typically contaminated with noise from surrounding waters rendering the state estimation process highly nonlinear. Using these measurements, the state estimation cannot be done with the traditional filters like Kalman filter[2] which works only for the linear dynamic systems. The estimated parameters in BOT usually diverge using the Kalman filter as the system and measurements used are nonlinearly related.

The foundational nonlinear algorithm available in the literature for BOT is extended Kalman filter (EKF). It gives effective results only when nonlinearity is of first order. The primary disadvantage of EKF is its flimsiness. The modified extended Kalman gain filter (MGEKF) is another alternative to EKF that improves the stability[3] in the estimation process convincingly. However, it gives only first order accuracy as EKF and fails in case of higher order of nonlinearity. The above mentioned problems are resolved by the unscented Kalman filter (UKF) that utilizes a sigma points approach to encapsulate the posterior covariance and mean, when propagated through the nonlinear system up to third-order accuracy[4,5]. EKF, MGEKF, and UKF use Gaussian assumption of noise in the estimation process and measurement models. Recent research work[6] done in this field, proposed filters like ensemble Kalman filter[7], shifted Rayleigh filter[8,9], Gauss-Hermite filter[10] and their variants also used Gaussian assumption of noise. The noise cannot be always Gaussian, this urges the requirement of nonlinear non Gaussian filter like particle filter[11] (PF) for state estimation.

The particle filter is a powerful Monte-Carlo approach to estimate a nonlinear non-Gaussian process. Being Monte-Carlo based recursive optimal Bayesian filtering technique, PF is neither restricted by linearity of system nor Gaussian distribution of state. The pivotal principle of PF is to collect random samples (particles) based on the representation of state's posterior density function, with similar weights and to calculate the estimates based on these samples with related

Received : 07 September 2020, Revised : 25 January 2021
Accepted : 18 February 2021, Online published : 01 July 2021

weights. These particles allow approximating any distribution of interest. This produce the finest state estimate as the number of particles reach infinity. Therefore, the accuracy of PF mainly depends on probability density function (pdf) of samples/particles and scheme of resampling of samples/particles that describe the system dynamics. Therefore, the research is centered on developing a good method of sampling density as well as refining the resampling scheme to increase performance of PF. The three main steps in PF are propagation of particles, particle weight calculation, and resampling. The first two steps of PF include particle dissemination and giving weights to the particles. The next step in PF is resampling which includes replacing the older set of particles with a new set of particles and associated weights generated based on the weights of the older particles[12,13].

The most important and crucial part is resampling[14] because it accounts for the generation of a new particles set with relevant weights in place of old ones. This resampling technique if not carried out properly, results in a corrupted set of particles causing sample impoverishment or sample degeneracy. By repeating the resampling steps, degeneracy of samples occurs when particles of less weight are eliminated, resulting in a situation where all the particles are concentrated at one point i.e., sample degeneracy that in turn results in sample impoverishment and vice-versa. Hence there is a necessity that proper care is taken while selecting the resampling scheme. Hence the research in choosing suitable resampling methods in PF has increased. There are many resampling techniques developed so far like random sampling, stratified, systematic, auxiliary, etc. However, there are some displeasing effects using resampling methods which urged the researchers to try other methods[15]. Besides, in realistic engineering applications, perturbations caused by irregularities in the measurement and kinematic noise model are inevitable. In addition, PF is also generating a costly computational charge due to the use of a large number of particles. Also, the environment in underwater is always unpredictable, making it difficult to select an optimal resampling scheme always. As proposed by Dan Simon[16], PF combination with other nonlinear filters like UKF called PFUKF filter is tried out for underwater BOT and the same is compared with filters like particle filter combined with other filters like EKF (PFEKF) and MGEKF (PFMGEKF). PFUKF is a method that uses unscented transform (UT) to compute the posterior state's probability density function for achieving improved sampling density function. Shared-memory architecture with parallel implementation, and reduction in order of the framework modelling help to minimize filtering dimensionality. This Improved algorithm structure is just a consideration that can be followed to boost PF's computational efficiency.

2. MATHEMATICAL MODELLING

2.1 System Modelling

In BOT, the target-observer scenario is modelled mathematically based on following assumptions. It is supposed that the observer and target are moving with constant course and speed. The target's state vector (X_s) in Cartesian coordinates of the target is represented as

$$X_s(\Gamma_{st}) = \left[\dot{r}_x(\Gamma_{st}) \quad \dot{r}_y(\Gamma_{st}) \quad r_x(\Gamma_{st}) \quad r_y(\Gamma_{st})\right] \tag{1}$$

where $\dot{r}_x(\Gamma_{st})$, $\dot{r}_y(\Gamma_{st})$, $r_x(\Gamma_{st})$, $r_y(\Gamma_{st})$ are the components of speed and range in the coordinates x and y, respectively at sample number Γ_{st}. The subsequent instant relative state vector based on the current instant $(X_s(\Gamma_{st}+1, \Gamma_{st}))$ is given by

$$X_s(\Gamma_{st}+1, \Gamma_{st}) = A(\Gamma_{st})X_s(\Gamma_{st}) + \mu\varepsilon(\Gamma_{st}) \tag{2}$$

where $A(\Gamma_{st})$ is matrix representing the system mobility given as

$$A(\Gamma_{st}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ t & 0 & 1 & 0 \\ 0 & t & 0 & 1 \end{bmatrix} \tag{3}$$

where 't' represents the interval at which samples are obtained.

It is assumed that the system noise $\varepsilon(\Gamma_{st})$, obey Gaussian distribution with mean=0 and covariance q and the system noise gain matrix μ is given in matrix form as

$$\mu = \begin{bmatrix} t & 0 \\ 0 & t \\ t^2/2 & 0 \\ 0 & t^2/2 \end{bmatrix} \tag{4}$$

The system noise covariance q is given in matrix form as

$$q(\Gamma_{st}) = \sigma^2 \begin{bmatrix} t^2 & 0 & t^3/2 & 0 \\ 0 & t^2 & 0 & t^3/2 \\ t^3/2 & 0 & t^4/4 & 0 \\ 0 & t^3/2 & 0 & t^4/4 \end{bmatrix} \tag{5}$$

where σ^2 is system noise variance.

The measurement equation is given by

$$b_m(\Gamma_{st}+1) = \tan^{-1}(r_x(\Gamma_{st})/r_x(\Gamma_{st})) + \gamma_b \tag{6}$$

where b_m is the measured bearing angle.

The measurement model equation Z at sample $\Gamma_{st}+1$ is specified by

$$Z(\Gamma_{st}+1) = H(\Gamma_{st}+1)X_s(\Gamma_{st}+1) + \gamma_b \tag{7}$$

where H is the matrix of the measuring model and γ_b is the noise of the measurement. Here the noise in the measurement is supposed to obey Gaussian distribution, with standard deviation of 0.33°.

The detailed mathematical modelling of PFEKF algorithm is given by Dan Simon[16] and for PFMGEKF algorithm refer[17].

2.2 PFUKF Algorithm

1. It is assumed that the initial state's pdf is known, based on which 'N' particles are randomly generated. The user selects the parameter 'N' as a trade-off between computational effort and precision estimation.

2. The state estimate (X_s) at time (Γ_{st}) based on all particles is calculated as

$$X_s(\Gamma_{st}) = \sum_{i=0}^{N} w^i(\Gamma_{st})\bar{x}_s^i(\Gamma_{st}) \tag{8}$$

where $i = 0,1,.........N$ particles, $\overline{x}_s^i(\Gamma_{st})$ is state vector for each particle, w^i is the weight associated with each particle, and the covariance, assuming independent and additive system noise, is computed as

$$\hat{P}(\Gamma_{st}) = \sum_{i=0}^{N} w^i(\Gamma_{st})\left[\overline{x}_s^i(\Gamma_{st}) - X_s(\Gamma_{st})\right] \\ *\left[\overline{x}_s^i(\Gamma_{st}) - X_s(\Gamma_{st})\right]^T \tag{9}$$

3. Prediction and updation of the covariance and mean of particles at time Γ_{st} using UKF.

(a) The sigma points of each particle are calculated as follows.

$$\overline{x}_s^i(\Gamma_{st}) = \begin{bmatrix} \overline{x}_s^i(\Gamma_{st}) \\ \overline{x}_s^i(\Gamma_{st}) + \left(\sqrt{(L_x+\lambda)\hat{P}(\Gamma_{st})}\right)_j \\ \overline{x}_s^i(\Gamma_{st}) - \left(\sqrt{(L_x+\lambda)\hat{P}(\Gamma_{st})}\right)_{j-L_x} \end{bmatrix}^T \tag{10}$$

where L_x is the state vector's dimension of target $\overline{x}_s^i$ of each particle. Also in the second term j varies as $j = 1,2,......L_x$ and in the third term j varies as $j = L_x + 1,......2L_x$. This is how the sigma points are generated for all the particles.

(b) Calculate the weights

$$W_0^{(M)} = \lambda / L_x + \lambda$$
$$W_0^{(c)} = \lambda / ((L_x+\lambda) + (1 - \vartheta^2 + \xi))$$
$$W_j^{(M)} = W_j^{(c)} = 1/(2(L_x+\lambda)) \;\; j = 1,2,.....2L_x \tag{11}$$

where λ is a scaling parameter given by $\lambda = \vartheta^2(L_x + \alpha) - L_x$. $\vartheta(=1e-3)$ is a small positive value that indicates the spread of sigma points around the mean. $\alpha(=0)$ is a secondary scaling parameter and ξ includes advance understanding of the distribution of x. $W_0^{(M)}$ and $W_0^{(c)}$ represent the target's state vector initial weights and weights of state covariance matrix. $W^{(M)}$ and $W^{(c)}$ represent the target's state sigma point vector and state sigma point covariance matrix respectively.

(c) The estimation of state predicted at time $\Gamma_{st}+1$ by using the measurement at Γ_{st} is computed as

$$\overline{x}_s^i(\Gamma_{st}+1,\Gamma_{st}) = \sum_{j=0}^{2L_x} W_j^{(M)} \overline{x}_s^i(j,(\Gamma_{st}+1,\Gamma_{st})) \tag{12}$$

and the covariance matrix predicted, assuming independent and additive system noise, is computed as

$$\hat{P}(\Gamma_{st}+1,\Gamma_{st}) = \sum_{j=0}^{2L_x} W_j^{(c)}[\overline{x}_s^i(j,(\Gamma_{st}+1,\Gamma_{st})) \\ -\overline{x}_s^i(\Gamma_{st}+1,\Gamma_{st})]*[\overline{x}_s^i(j,(\Gamma_{st}+1,\Gamma_{st})) - \overline{x}_s^i(\Gamma_{st}+1,\Gamma_{st})]^T \tag{13}$$

Using the measurement facsimile given in eqn. (5), transform the expected sigma points.

(d) The measurement model matrix prediction $\hat{Z}(\Gamma_{st}+1,\Gamma_{st})$ is given as

$$\hat{Z}(\Gamma_{st}+1,\Gamma_{st}) = \sum_{j=0}^{2L_x} W_j^{(M)} Z(j,(\Gamma_{st}+1,\Gamma_{st})) \tag{14}$$

where $Z(j,\Gamma_{st}+1,\Gamma_{st}) = H(\overline{x}_s^i(\Gamma_{st}+1,\Gamma_{st}))$

(e) The computation of innovation covariance matrix $\hat{P}_{zz}$ is given as

$$\hat{P}_{zz}(\Gamma_{st}+1,\Gamma_{st}) = \sum_{j=0}^{2L_x} W_j^{(c)}\left[Z(j,(\Gamma_{st}+1,\Gamma_{st})) - \hat{Z}(\Gamma_{st}+1,\Gamma_{st})\right] \\ *\left[Z(j,(\Gamma_{st}+1,\Gamma_{st})) - \hat{Z}(\Gamma_{st}+1,\Gamma_{st})\right]^T + \sigma_b^2(\Gamma_{st}) \tag{15}$$

(f) The computation of cross covariance matrix $\hat{P}_{xz}$ is given as

$$\hat{P}_{xz}(\Gamma_{st}+1,\Gamma_{st}) = \sum_{j=0}^{2L_x} W_j^{(c)} \\ *\left[\overline{x}_s^{\,i}(j,(\Gamma_{st}+1,\Gamma_{st})) - \overline{x}_s^{\,i}(\Gamma_{st}+1,\Gamma_{st})\right] \\ *\left[Z(j,(\Gamma_{st}+1,\Gamma_{st})) - (\hat{Z}(\Gamma_{st}+1,\Gamma_{st})\right]^T \tag{16}$$

(g) The computation of Kalman gain G^i is given as

$$G^i(\Gamma_{st}+1) = \hat{P}_{xz}(\Gamma_{st}+1,\Gamma_{st})(\hat{P}_{zz}(\Gamma_{st}+1,\Gamma_{st}))^{-1} \tag{17}$$

(h) The computation of estimated state $\overline{x}_s^{\,i}$ is given as

$$\overline{x}_s^{\,i}(\Gamma_{st}+1) = \overline{x}_s^{\,i}(\Gamma_{st}+1,\Gamma_{st}) \\ +G^i(\Gamma_{st}+1)(Z(\Gamma_{st}+1) - \hat{Z}(\Gamma_{st}+1,\Gamma_{st})) \tag{18}$$

where $Z(\Gamma_{st}+1)$ is measurement vector matrix.

(i) The estimation of the error covariance matrix is computed as

$$\hat{P}(\Gamma_{st}+1) = \hat{P}(\Gamma_{st}+1,\Gamma_{st}) \\ -G^i(\Gamma_{st}+1)\hat{P}_{zz}(\Gamma_{st}+1,\Gamma_{st})(G^i(\Gamma_{st}+1))^T \tag{19}$$

4. Now calculate the relative likelihood Q_i of each particle $\overline{x}_s^{\,i}(\Gamma_{st}+1)$ conditioned on measurement.

5. Scale the relative probability calculated in step 4 as follows:

$$w^i(\Gamma_{st}+1) * w^i(\Gamma_{st}) \\ *probabilityof(\overline{x}_s^{\,i}(\Gamma_{st}+1) \mid \overline{x}_s^{\,i}(\Gamma_{st})) \\ *probabilityof(Z(\Gamma_{st}+1) \mid \overline{x}_s^{\,i}(\Gamma_{st})) \tag{20}$$

and normalize the weights using eq. (21)

$$\sum_{i=1}^{N} \overline{w}^i(\Gamma_{st}+1) = 1 \tag{21}$$

This redefines the set of a posteriori particles $\overline{x}_s^{\,i}(\Gamma_{st})$ and covariance $\hat{P}(\Gamma_{st})$ based on the relative likelihoods $w^i(\Gamma_{st})$.

6. Posteriori particles $\overline{x}_s^{\,i}(\Gamma_{st})$ and covariance $\hat{P}(\Gamma_{st})$ are obtained now. Calculate the target state parameters. On the income of the next measurement set $\Gamma_{st} = \Gamma_{st}+1$ and go to step 3 otherwise exit.

The flowchart of the PFUKF algorithm is given in appendix A.

3. SIMULATION AND RESULTS

To implement the proposed algorithm, the initialisations are done as follows. The observer is chosen to be at origin and the target is chosen to be at some position P. The speed of observer is assumed to be less than target speed which helps the observer to track the target easily. Observer course, target course and the bearing measurements are measured w.r.t true north. Initial bearing is taken in between 0-40°. The observer performs the traditional S- maneuver in the line of sight (LOS) as given in figure 1 which helps to get the observability of the process. Initially the particles are generated randomly using 'randn'. For each particle, UKF algorithm is applied. The initial estimate of target state vector is chosen as

$$X_s(0,0) = \begin{bmatrix} 5 & 5 & 5000\sin b_m & 5000\cos b_m \end{bmatrix} \tag{22}$$

The particles generated are added to the state vector initial estimate. The elements of initial covariance diagonal matrix which follows uniform distribution is given as

$$\hat{P}(0,0) = diagonal\begin{bmatrix} \frac{4\dot{r}_x^2(0,0)}{12} & \frac{4\dot{r}_y^2(0,0)}{12} & \frac{4r_x^2(0,0)}{12} & \frac{4r_y^2(0,0)}{12} \end{bmatrix} \tag{23}$$

The number of particles taken for simulation purpose is 1000. This number can be increased further but this enhances both complexity and computation time. The application and accuracy required decides the number of samples/particles to be selected. As the BOT is a stochastic process, it is necessary to calculate the confidence of the solution obtained.

It is readily done by PF, is Monte-Carlo based simulation method. Every filter is simulated for 800 measurement samples. The scenarios chosen for simulation, following above-mentioned criteria, are given in Table 1. The scenarios are categorised depending upon angle on target bow (ATB)[5] as low, medium and high. When the ATB is in the range of 0°-30°, it is termed as low ATB. Similarly, the range of angles 30°-40° and 40°-90° are termed as medium and high ATB respectively.

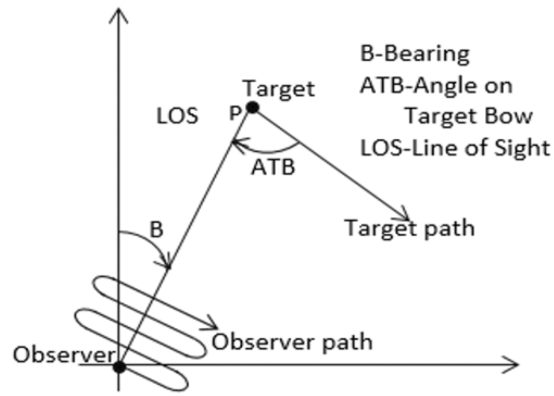


Figure 1. Observer and target movements.

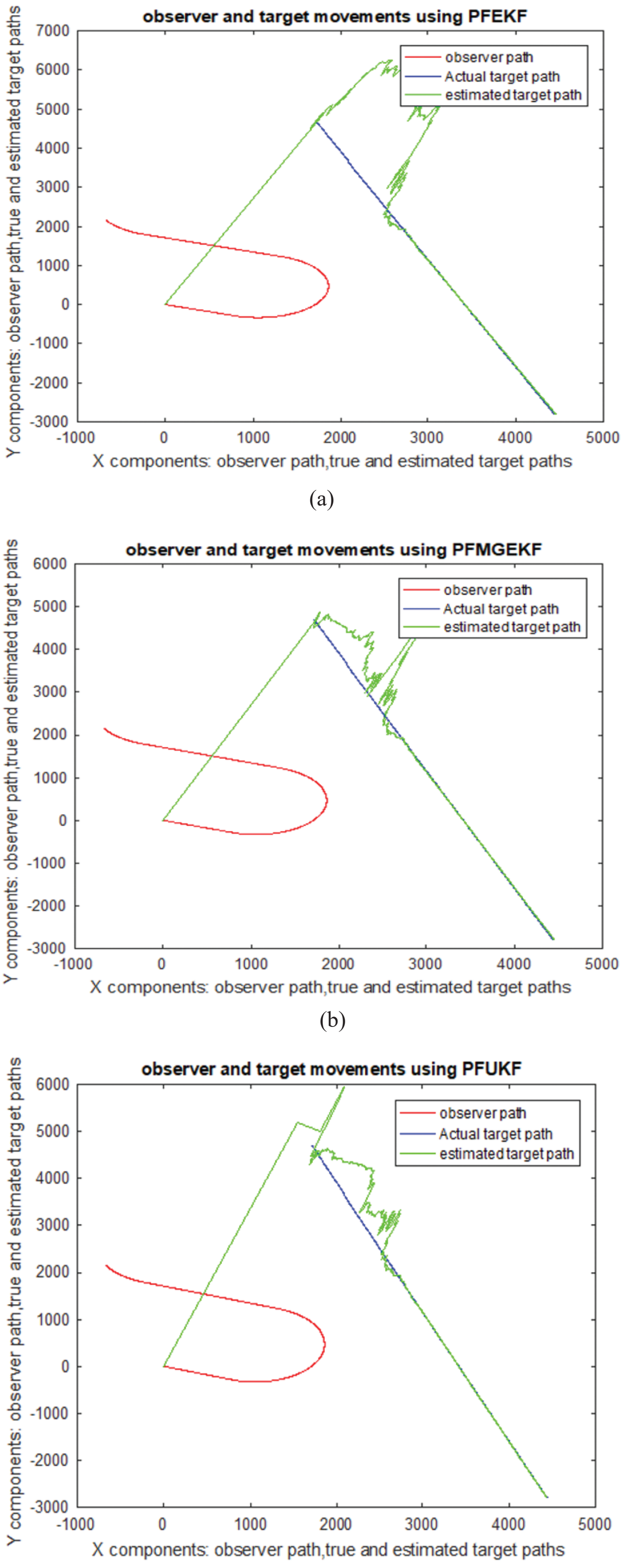
Depending on the guidance algorithm of weapon, the solution is said to be achieved when the Root Mean Square (RMS) error in the range estimate is less than 2.66% of the true range, the RMS error in the course estimate is less than 1° and the RMS error in the speed estimate is less than 0.33 m/s. These scenarios are evaluated in the Matlab platform. Based on the acceptance criteria, the solution obtained for the scenarios in Table 1 using PFEKF, PFMGEKF and PFUKF is tabulated in Table 2. By observing the results, it is understood that using PFUKF, solution is obtained in only medium ATB scenarios whereas PFEKF and PFMGEKF gives solution for all scenarios. It is emphasised from Table 2 that PFEKF and PFMGEKF are generating solution for all scenarios and the convergence time obtained is almost similar for all scenarios. MGEKF is more stable than EKF however when combined with PF, stability in the filtering algorithm is reduced. PFUKF fails for many scenarios because of the complexity that is created in execution by combining with UKF. The UKF uses sigma points to estimate the state mean and state covariance. When UKF is used alongside PF which already uses randomly generated particles for state estimation, the complexity multiplies and creates hurdles to state estimation rather than improving it. The target path estimated using PFEKF, PFMGEKF and PFUKF for scenario 5 (medium ATB) and scenario 7 (high ATB) are given in figures 2 and 3 respectively. The true and estimated paths of the target are

Table 1. Scenarios selected for performance analysis of the filters

Scenario no.	Range (m)	Observer's velocity (m/s)	Target's velocity (m/s)	Target's initial course (deg)	Initial bearing (deg)	ATB (deg)
1	4000	8	12	180	10	Low
2	4000	8	12	185	30	
3	4500	9	11	170	10	
4	4500	9	11	165	20	Medium
5	5000	7	10	160	20	
6	5000	7	10	170	20	
7	4000	8	12	135	30	High
8	4500	9	11	160	40	
9	5000	7	10	145	20	

Table 2. Convergence time (CT) in seconds for 1000 particles

s.no	PFEKF				PFMGEKF				PFUKF			
	Range	Speed	Course	CT	Range	Speed	Course	CT	Range	Speed	Course	CT
1	291	291	288	291	289	289	283	289	-	-	452	-
2	264	298	274	298	299	270	298	299	639	-	356	-
3	393	366	377	393	382	366	377	382	-	-	-	-
4	367	288	348	367	324	268	292	324	374	320	337	374
5	289	347	382	382	289	347	382	382	288	347	348	348
6	347	256	242	347	347	275	249	347	366	417	361	417
7	270	347	458	458	382	382	466	466	-	-	-	-
8	256	303	432	432	298	320	438	438	491	-	506	-
9	259	354	432	432	320	382	438	438	338	-	441	-

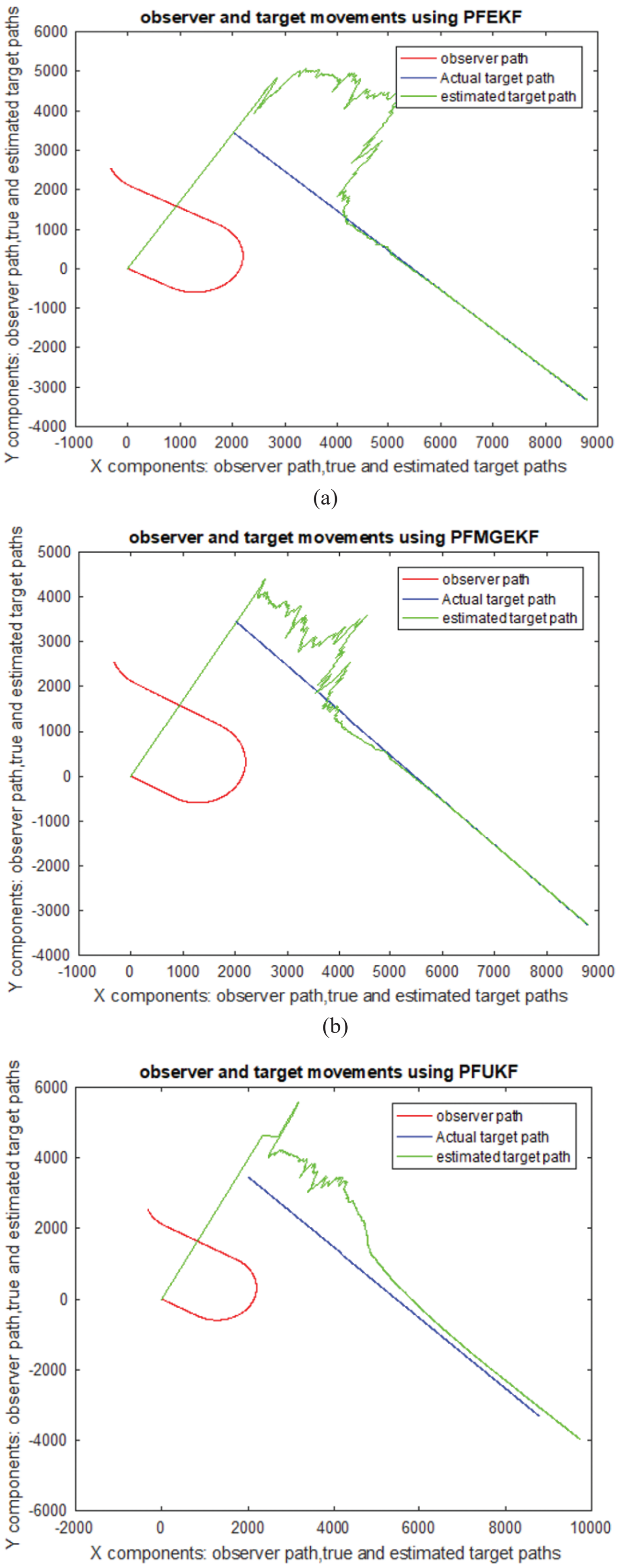


(a)

(b)

(c)

Figure 2. (a) Observer, target's true and estimated paths using PFEKF for scenario 5, (b) Observer, target's true and estimated paths using PFMGEKF for scenario 5, and (c) Observer, target's s true and estimated paths using PFUKF for scenario 5.



(a)

(b)

(c)

Figure 3. (a) Observer, target's true and Estimated paths using PFEKF for scenario 7, (b) Observer, target 's true and estimated paths using PFMGEKF for scenario 7, and (c) Observer, target's true and estimated paths using PFUKF for scenario.

converged using PFEKF, PFMGEKF and PFUKF algorithms for scenario 5 and the same is shown in figure 2(a) to 2(c). The true and estimated paths of the target are diverged using PFEKF, PFMGEKF and PFUKF algorithms for scenario 7 and the same is shown in figure 3(a) to 3(c).

From Table 2, it is understood that for medium ATB scenarios, all the three algorithms proved to be efficient whereas PFUKF failed for low and high ATB scenarios. For a sample scenario say 4, PFEKF converges at 367, 288 and 348 seconds in range, speed and course respectively and the overall convergence time is 367 seconds. For the same scenario, PFMGEKF converges at 324, 268 and 292 seconds in range, speed and course respectively and here, the overall convergence time is 324 seconds. Similarly, PFUKF converges at 374, 320 and 337 seconds in range, speed and course respectively for the same scenario 4 and here, the overall convergence time is 374 seconds. Similarly, for scenario 2, the overall convergence time using PFEKF and PFMGEKF are 298 and 299 seconds respectively. Using PFUKF, only the range and course converged at 639 and 356 seconds respectively and the speed is not converged. The RMS error in estimated range, speed and course of target using PFEKF, PFMGEKF and PFUKF algorithms for scenarios 5 and 7 are given in figures 4(a) to 4(c) and 5(a) to 5(c) respectively. From figures 4(a) and 5(a), it is understood that estimated range's RMS error using PFEKF is less when compared to PFMGEKF and PFUKF. The same is observed in comparison of RMS error of course and speed using the algorithms which given by figures 4(b) and 5(b) for speed, figures 4(c) and 5(c) for course. Also, PFMGEKF is having less RMS error than PFUKF i.e. in low and high ATB scenarios, PFUKF diverges fast and then PFMGEKF diverges.

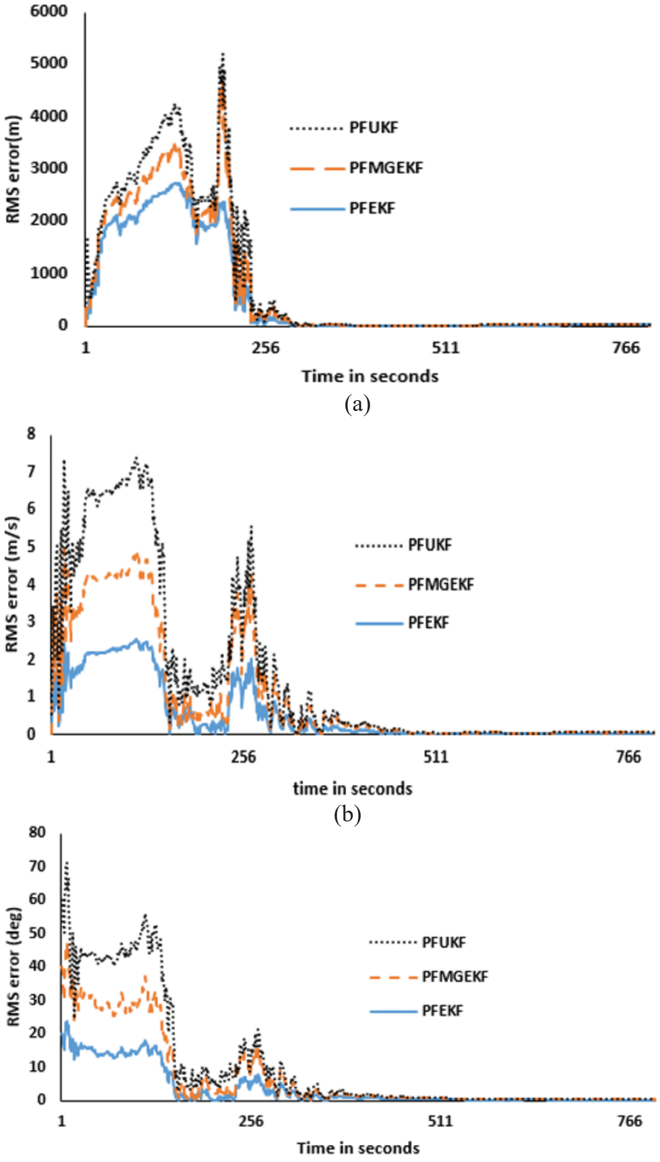


Figure 4. (a) Estimated range's RMS error for scenario 5, (b) Estimated speed's RMS error for scenario 5, and (c) Estimated course's RMS error for scenario 5.

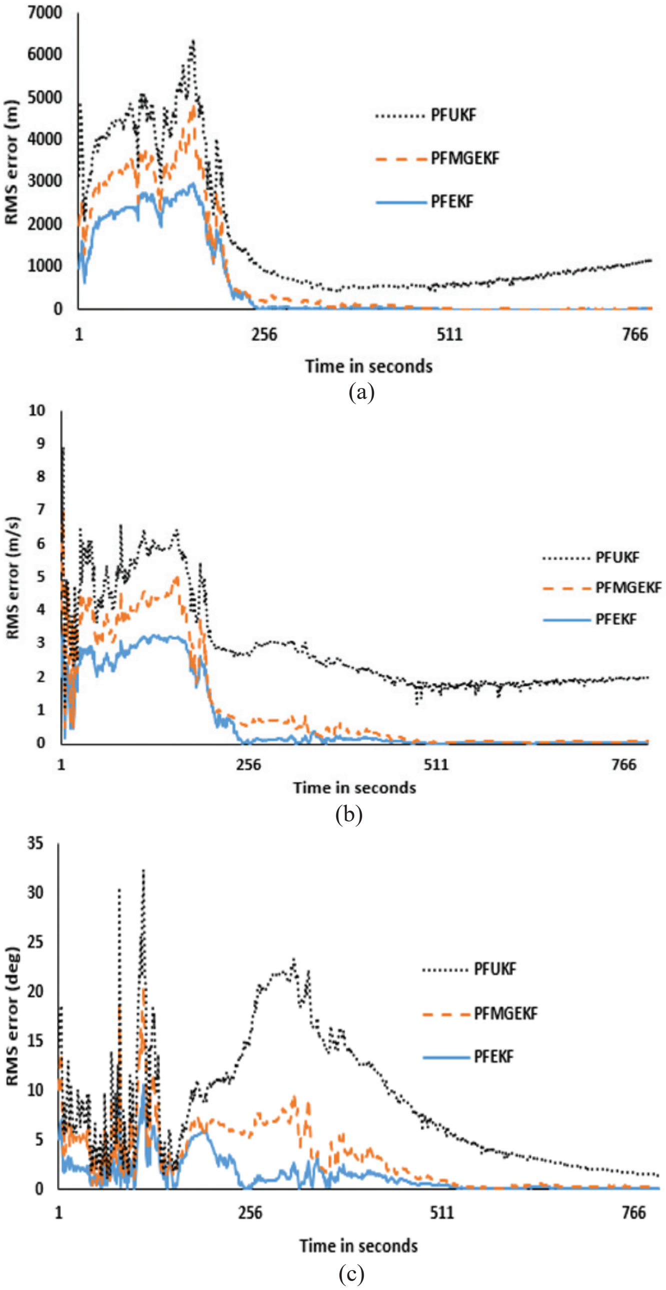


Figure 5. (a) Estimated range's RMS error for scenario 7, (b) Estimated speed's RMS error for scenario 7, and (c) Estimated course's RMS error for scenario 7.

Hence it is suggested to utilize PFEKF for estimating the TMP in BOT and obtaining the solution with faster convergence. The efficiency of PFEKF and PFMGEKF is nearly equal for all the scenarios and PFUKF failed for low and high ATB scenarios.

4. CONCLUSION

This paper analyses the application of PFUKF algorithm and compares the performance with PFEKF, PFMGEKF for underwater state estimation using bearings only measurements. The algorithms are simulated using Matlab. From the analysis of the simulated results, it is evident that the PFUKF filter is gives weaker results when compared to other PF combinations. UKF's sigma points approach for state estimation enables it to give a solution to third-order accuracy in any nonlinear process. PF uses samples/particles to represent the real pdf of the target state. However, the combination of the two i.e. PF and UKF in PFUKF, generates 9 sigma points for each particle thereby multiplying the computational complexity. Hence PFUKF instead of giving a very good solution fails to give a solution. MGEKF stabilizes the solution and also prevents the divergence of the solution once it converges. However, this is not the case in PFMGEKF, the modified gain added to EKF for stability purpose, increase the complexity further when combined with PF. Hence this paper recommends the utilisation of PFEKF for underwater state estimation rather than the proposed algorithm PFUKF.

REFERENCES

1. Lindgren, A. G. & Gong , K. F. Position and velocity estimation via bearing observations. *IEEE Trans. Aerospace and Electronic Systems,* 1978, **14**(4), 564-577.
   doi: 10.1109/TAES.1978.308681
2. Aidala, V. J. Kalman filter behaviour in bearings-only tracking applications, *IEEE Trans. Aerosp. Electron. Syst.*, 1979, **15**(1), 29-39.
   doi: 10.1109/TAES.1979.308793
3. Song, T. L. & Speyer, J. L. A stochastic analysis of a modified gain extended Kalman filter with applications to estimation with bearings only measurements. *IEEE Trans. Autom. Control,* 1985, **30(**10), 940-949.
   doi: 10.1109/CDC.1983.269736
4. Julier , S. J. & Uhlmann , J. K. A new extension of the Kalman filter to nonlinear systems. *In* I*nternational Symposium on Aerospace / Defence Sensing, Simulation and Control*, 1997, 182-193.
   doi: 10.1117/12.280797
5. Koteswara Rao, S. Bearings-Only Tracking: Observer Maneuver Recommendation. *IETE Journal of Research,* 2018, **67**(2), 193-204.
   doi: 10.1080/03772063.2018.1535917
6. Budhiraja, A.; Chen, L. & Lee, C. A survey of numerical methods for nonlinear filtering problems. P*hysica D: Nonlinear Phenomena*, 2007, **230**(1–2), 27-36,
   doi: 10.1016/j.physd.2006.08.015
7. Sun, T. & Xin, M. Bearings-Only Tracking Using Augmented Ensemble Kalman Filter. *IEEE Transactions on Control Systems Technology*, 2020, **28**(3), 1009–1016.
   doi: 10.1109/TCST.2018.2890370
8. Hou, J.; Yang, Y. & Gao, T. Variational Bayesian Based Adaptive Shifted Rayleigh Filter for Bearings-Only Tracking in Clutters. *Sensors,* 2019, **19**(7).
   doi: 10.3390/s19071512
9. Sharma, N.; Tiwari, R. & Bhaumik, S. Risk Sensitive Shifted Rayleigh Filter for Underwater Bearings-Only Target Tracking Problems, 1st Maritime Situational Awareness Workshop (MSAW), Italy, 2019.
   doi: 10.13140/RG.2.2.15130.06082
10. Singh, A.K.; Radhakrishnan, R.; Bhaumik, S. & Date, P. Adaptive Sparse-grid Gauss-Hermite Filter. *Journal of Computational and Applied Mathematics,* 2018, **342**, 305-316.
    doi: 10.1016/j.cam.2018.04.006
11. Gordon, N. J.; Salmond, D. J. & Smith, A. F. M. Novel approach to nonlinear/non-Gaussian Bayesian state estimation, 1993. *In* IEE Proceedings F - Radar and Signal Processing, **140**(2),107-113.
    doi: 10.1049/ip-f-2.1993.0015
12. Arulampalam, M. S.; Maskell, S.; Gordon, N. & Clapp, T. A tutorial on particle filters for online nonlinear/ non-Gaussian Bayesian tracking, *IEEE Trans. Signal Processing*, 2002, **50**(2), 174–188.
    doi: 10.1109/78.978374
13. Cappe, O.; Godsill, S. J. & Moulines, E. An overview of existing methods and recent advances in sequential Monte Carlo. *IEEE Proc.*, 2007, **95**(5) , 899–924.
    doi: 10.1109/JPROC.2007.893250
14. Li, T. ; Bolic, M. & Djuric, P.M. Resampling methods for particle Filtering: Classification, implementation and strategies. I*EEE Signal Processing Magazine,* 2015, **32**(3), 70-86.
    doi: 10.1109/MSP.2014.2330626
15. Wang, X. & Ni, W. An improved particle filter and its application to an INS/GPS integrated navigation system in a serious noisy scenario. *Measurement Science and Technology*, 2016, **27**(9).
    doi: 10.1088/0957-0233/27/9/095005
16. Dan Simon, Optimal State Estimation: Kalman, H∞ and Nonlinear Approximations, Wiley, 2006.
17. Jahan, K. & Rao, S.,K. Fusion of Angle Measurements from Hull Mounted and Towed Array Sensors. *Information*, 2020, **11**(9), 432.

CONTRIBUTORS

Ms G. Naga Divya, graduated from Koneru Lakshmaiah College of Engineering, Acharya Nagarjuna University, Guntur in Electronics and Communication Engineering in 2010 and earned a Master's degree in Radar and Microwave Engineering from Andhra University in 2014. (Deemed to be university). In the present research, she carried out mathematical modelling and realisation of the algorithm using Matlab.

Dr S. Koteswara Rao, former "G" scientist in NSTL, DRDO, Visakhapatnam, is presently employed as a professor at the Vaddeswaram, Guntur, AP, India, Koneru Lakshmaiah Education Foundation. In 1977, he obtained a BTech degree from JNTU and ME in 1979 from PSG Technology College, Coimbatore, and a Ph.D. in Electrical Engineering from Andhra University. He published numerous papers in the field of stochastic signal processing in the International Conference & Journals. He is a fellow IETE member.

In the present research, he verified the mathematical modelling, code and the results. He helped in the completion of the manuscript.

Appendix A

The state and covariance of the target's state are initialised. N particles are randomly created based on the pdf of the target's state at time Γ_{st}. For each particle, UKF algorithm is applied to generate the new particle using the measurement $Z(\Gamma_{st})$. The mean and covariance are calculated for each particle at time Γ_{st}. The weights are updated using the relative likelihood. Now the weights are normalised and new particles' set and covariance is created. The mean of all the particles is calculated to get a single value for each element of target state vector. Now the target motion parameters are estimated. The sample number is incremented to the process is repeated to update the state using the incoming measurement.

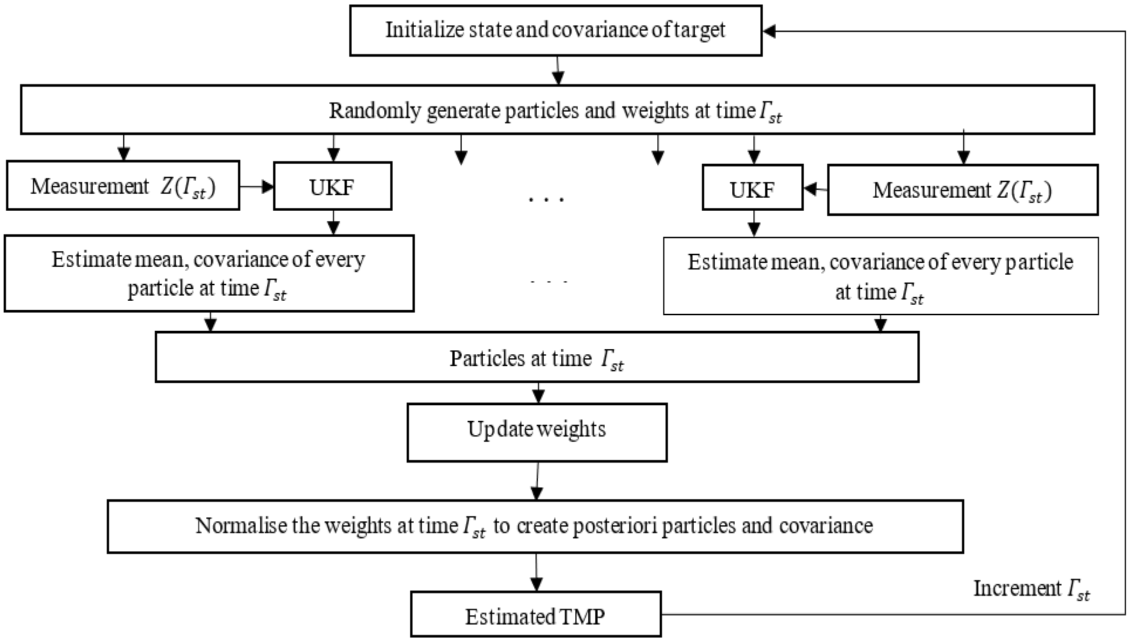

Initialize state and covariance of target
Randomly generate particles and weights at time Γ_{st}
Measurement $Z(\Gamma_{st})$
UKF
. . .
UKF
Measurement $Z(\Gamma_{st})$
Estimate mean, covariance of every particle at time Γ_{st}
. . .
Estimate mean, covariance of every particle at time Γ_{st}
Particles at time Γ_{st}
Update weights
Normalise the weights at time Γ_{st} to create posteriori particles and covariance
Estimated TMP
Increment Γ_{st}

Figure A. Flowchart of PFUKF algorithm.