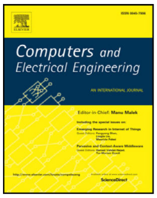
Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng



Implementation Of underwater target tracking techniques for Gaussian and non-Gaussian environments[☆]

Kausar Jahan[*], S. Koteswara Rao

Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, A.P, India

ARTICLE INFO

Article history:
Received 15 December 2019
Revised 12 June 2020
Accepted 15 July 2020

Keywords:
Cramer-rao lower bound
Extended kalman filter
Gaussian mixture noise
Shot noise
Target tracking
Unscented kalman filter

ABSTRACT

There are several techniques implemented, in an underwater target tracking environment, for the nonlinear dynamic systems in Gaussian and non-Gaussian environments. It is assumed with non-Gaussian distribution to make the problem part of the non-Gaussian distribution, and is measured in terms of calculations of plenty of scenarios simulated to validate the potential of the sub-optimal filter.This research is further carried out by considering two categories of non-Gaussian noises i.e.a mixture of Gaussian noises and shot noise. To evaluate tracking in Gaussian and non-Gaussian noises, the suboptimal filters, Extended Kalman filter, and Unscented Kalman filter (UKF algorithms are considered. Gaussian noise is a statistical noise having probability density function equal to the normal distribution function**.** The suboptimal filters, Extended Kalman filter, and Unscented Kalman filter (UKF) algorithms are considered to evaluate tracking in Gaussian and non-Gaussian noises. To make further evaluation of the above said algorithms, they are compared with theoretical Cramer-Rao lower bound. The efficiency of UKF is in terms of percentage of non-Gaussian noise corrupted measurements, for which solution is obtained within a short time. The application of Monte-Carlo method at this simulations trapped accurate results.

1. Introduction

1.1. Background

The usage of techniques by using bearing measurements for tracking Underwater has decades of history. They are also applied to surveillance of enemyships and submarines. Tracking techniques are most useful in anti-submarine warfare when observers are under attack.

Bearing measurement is the angle measured from a reference axis to the line joining target and observer. Tracking with the help of only bearing measurements, popularly known as Bearings-Only Tracking (BOT), has a wide range of applications. BOT is being studied extensively since the 1980s [1–2]. The analysis of the movement of the target like speed, course, and range is termed as Target Motion Analysis, popularly known as TMA.

The justification of the work is to find kinematics of target with available measurements, which are corrupted with non-Gaussian noises. The scenario consisting of single sonar mounted on the observer, operating in passive mode, is used for

[☆] This paper is for CAEE special section VSI-mis. Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. Chan-Yun Yang.

[*] Corresponding author.
E-mail address: kausar.465@gmail.com (K. Jahan).

https://doi.org/10.1016/j.compeleceng.2020.106783
0045-7906/© 2020 Elsevier Ltd. All rights reserved.

tracking the target. The low risk has recognized by following others when the sonar operation in passive mode. The TMA of the target moving with incessant velocity has been studied and discussed in the literature.The bearing observations are tangentially bonded to the target state i.e., range of the target, making the process nonlinear. The nonlinear relation between measurement and target state is mathematically given in Section 2.1. As the process of tracking is non-linear, the observer following a constant velocity and straight path makes the process unobservable. So, S-manoeuvre is used for observing the dynamic behavior of the enemy model so that the process becomes more observable. These observations of the process in underwater scenarios are discussed in [3–5]. The solution to the above problem was obtained by using conventional methods like Maximum Likelihood estimator, Pseudo Linear estimator, least squares estimators, etc and by using modern methods from Kalman filter to spare-grid Gauss-Hermite filter.

For the EKF, the state initializations should be more precise and the noise in state and measurements should be following Gaussian distribution. As EKF is a suboptimal filter, the solution diverges as the nonlinearity and the noise increases. Other filters like Unscented Kalman Filter (UKF) [6–8], Particle filter (PF), cubature Kalman filter (CKF), hybrid filters, etc. [9–15] approximates the target state and their probability densities using deterministic resampling method.

1.2. Literature

In the real-time application of target tracking, the process noise and the measurement noise will not follow Gaussian distribution. So, the filtering algorithms designed based on just the Gaussian noise model may not perform well in real-time tracking problems. The solution to the non-Gaussian noise models includes PF [16] which depends on Monte-Carlo integration and sampling techniques and Gaussian sum filter [17,18] that depends on the mixing up of different Gaussian distributions with different variances. In PF, the need for the assumption of probability density functions is eliminated but the computational effort increase as the number of particles increases for the sake of accuracy in estimated solution. Hence research is being carried [19,20] to reduce the disadvantages in PF due to sample impoverishment and degeneracy. Another filter that shows a comparable functioning to that of PF in passive target tracking is the Gaussian sum cubature Kalman filter [21]. Other filters include improved Gaussian mixture filter algorithm [22], and limited Gaussian mixture model [23]. The weight of the Gaussian components is assumed to be persistent in all the above techniques while communicating the uncertainty through the non-linear system. These are refurbished at the measurement update phase only. This premise is viable when the measurements are available continuously or have low noise levels and the system has minute nonlinearity. The same can't be applied for non-linear cases in real-time. A new Gaussian sum filter, that makes use of weights of the Gaussian mixture model in both measurement and time update phases, was proposed by Terejanu [23].

1.3. Motivation

All the suboptimal filters have been suggested as solutions for the TMA under Gaussian noise conditions. Filters like particle filter and shifted Rayleigh filter [11] were suggested for filtering non-linear systems under the non-Gaussian noise conditions but have high computational complexities. To perform highly complicated computations, more memory is required and needs a high-speed processor. So, work is carried out to find the ability of the suboptimal filters (EKF and UKF, that have low computational complexity) under non-Gaussian noise assuming that the non-Gaussian noise prevails only at certain time samples and not throughout the process.

The estimated target parameters are erroneous as the system model and measurements are corrupted with noise. These errors are reduced using different filtering algorithms. For the weapon to be fired on to the target, these errors must be low, and the errors must be reduced in less time.In surveillance applications, accuracy in estimated parameters must be obtained in a short period. So, the suboptimal filters with less complexity in calculations are more advantageous than complex nonlinear filters.

Thebehavior of suboptimal filters, UKF, and EKF are evaluated in the shot noise (SN) and Gaussian mixture noise (GMN) environments. The target is presumed to be exhibit constant velocity motion. The mathematical modeling of the algorithms is discussed in detail in Section 2. The scenarios on different Angle on Target Bow (ATB) are considered for simulation. ATB is the discrepancy between the target course and the line of sight. The target encountering the observer at high (41° – 90°), medium (31° – 40°), and low (0° – 30°) ATB scenarios are considered to assess the algorithms. The target moves away from the observer,with ATB greater than 90°, and is therefore of no worth in tracking. The simulation and results obtained using MATLAB are mentioned in Section 3. The overall summary of the work done is given in Section 4.

2. Mathematical modeling

2.1. Target motion analysis

Consider the observer is at position 'O' initially and the target movement is assumed to be straight-line keeping the speed at a constant value. The state vector at time instant 'τ' of the observer [1] is represented as in Eq. (1)

$$S_o(\tau) = \begin{bmatrix} \nu_{xo}(\tau) & \nu_{yo}(\tau) & r_{xo}(\tau) & r_{yo}(\tau) \end{bmatrix}^T \tag{1}$$

where$v_{xo}(\tau)$, $v_{yo}(\tau)$, $r_{xo}(\tau)$, $r_{yo}(\tau)$ represents the speed and range of the observer in x and y-direction. The change in the observer position is obtained from its course and speed as in Eq. (2) and Eq. (3)

$$dr_{xo}(\tau) = v_{xo}(\tau) * \sin ocr * t \tag{2}$$

$$dr_{yo}(\tau) = v_{yo}(\tau) * \cos ocr * t \tag{3}$$

where $dr_{xo}(\tau), dr_{yo}(\tau)$are the change in x-coordinate and y-coordinates of observer and ocr is the observer course angle and t is the time of one second [1]. The relative state vector [1,3] of the target is represented as in Eq. (4)

$$S_s(\tau) = \begin{bmatrix} v_x(\tau) & v_y(\tau) & r_x(\tau) & r_y(\tau) \end{bmatrix}^T \tag{4}$$

where$v_x(\tau)$, $v_y(\tau)$, $r_x(\tau)$, $r_y(\tau)$ are relative components of speed and range in x and y coordinates respectively. The relative state vector for the next time based on the present time state vector is calculated as in Eq. (5)

$$S_s(\tau+1) = A(\tau)S_s(\tau) + b(\tau+1) + \omega C(\tau) \tag{5}$$

where $A(\tau)$ is the system dynamics matrix calculated as in Eq. (6)

$$A(\tau) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ t & 0 & 1 & 0 \\ 0 & t & 0 & 1 \end{bmatrix} \tag{6}$$

$C(\tau)$ is the process noise and ω is calculated as in Eq. (7)

$$\omega = \begin{bmatrix} t & 0 \\ 0 & t \\ t^2/2 & 0 \\ 0 & t^2/2 \end{bmatrix} \tag{7}$$

$b(\tau)$ is a deterministic matrix and is calculated as in Eq. (8)

$$b(\tau) = \begin{bmatrix} 0 \\ 0 \\ -(r_{xo}(\tau) - r_{xo}(\tau-1)) \\ -(r_{yo}(\tau) - r_{yo}(\tau-1)) \end{bmatrix}^T \tag{8}$$

The covariance of the process noise is calculated as in Eq. (9)

$$Q(\tau) = E\left[(\omega C(\tau))(\omega C(\tau))^T\right]$$
$$Q(\tau) = \sigma^2 \begin{bmatrix} t^2 & 0 & t^3/2 & 0 \\ 0 & t^2 & 0 & t^3/2 \\ t^3/2 & 0 & t^4/4 & 0 \\ 0 & t^3/2 & 0 & t^4/4 \end{bmatrix} \tag{9}$$

where σ^2 represents the variance in the process noise.

The measurement equation for this application has only bearing angles and the bearing angle $\beta(\tau)$ is represented as in Eq. (10)

$$\beta_m(\tau) = \tan^{-1}(r_x(\tau)/r_y(\tau)) + \Upsilon_B \tag{10}$$

where Υ_B is the measurement noise, assumed to be following Gaussian distribution with σ_B^2variance.

2.2. UKF Algorithm

UKF is straight forward add-on of the Unscented Transformation (UT) to the recursive estimation. In UKF, the concatenation of the original states and noise variables are delineated as the state random variables. The sigma point selection method of UT is implemented to the delineated state random variables to calculate the corresponding matrix of sigma points. The weighted mean and covariance of posterior sigma points are utilized to estimate the mean and covariance of the state of the target. [7,8]. The flowchart showing the implementation of the UKF algorithm is as shown in Fig. 1 and the implementation steps are as follows:

a) Let L_1 be the dimension of the target state vector. $(2L_1 + 1)$ state vectors are calculated from the initial points using sigma points as in Eq. (11)

$$S(\tau) = \begin{bmatrix} S_s(\tau) \\ S_s(\tau) + \sqrt{(L_1+\lambda) + P(\tau)} \\ S_s(\tau) - \sqrt{(L_1+\lambda) + P(\tau)} \end{bmatrix}^T \tag{11}$$

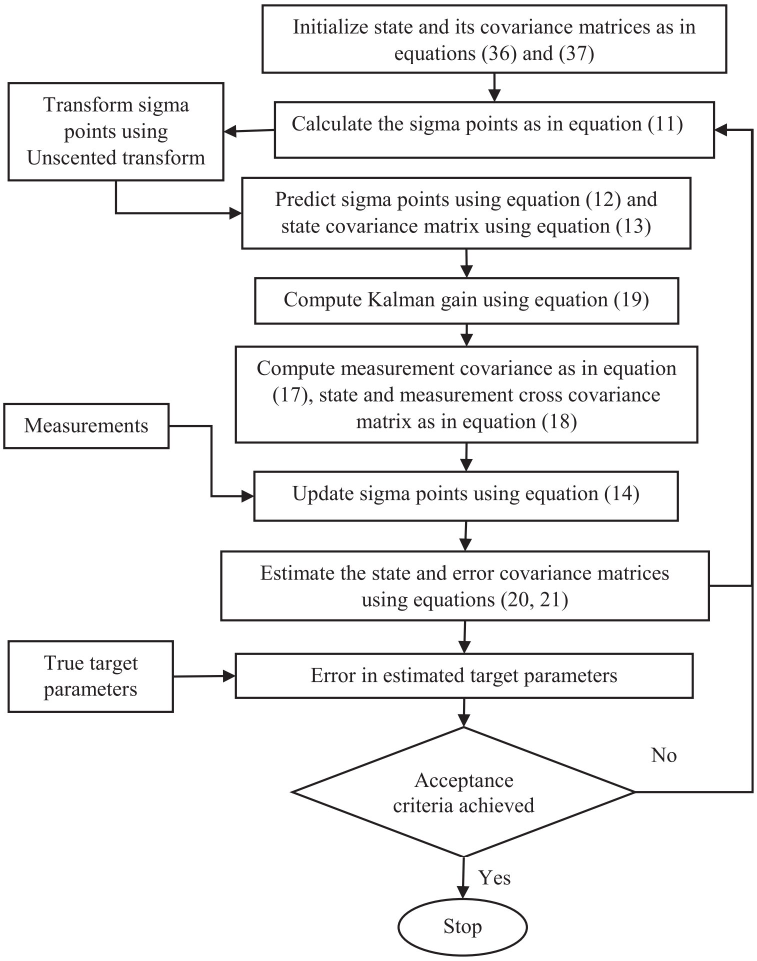

Initialize state and its covariance matrices as in equations (36) and (37)
Calculate the sigma points as in equation (11)
Transform sigma points using Unscented transform
Predict sigma points using equation (12) and state covariance matrix using equation (13)
Compute Kalman gain using equation (19)
Compute measurement covariance as in equation (17), state and measurement cross covariance matrix as in equation (18)
Measurements
Update sigma points using equation (14)
Estimate the state and error covariance matrices using equations (20, 21)
True target parameters
Error in estimated target parameters
Acceptance criteria achieved
No
Yes
Stop

Fig. 1. Flow chart representing UKF algorithm

Here λ is a scaling parameter and is calculated as follows.

$$\lambda = \vartheta^2(L_1+\alpha) - L_1$$

ϑ is set to a small positive value (e.g., 1e-3) that determines how the mean is surrounded by the sigma points. α, set to two, is a secondary scaling parameter as defined in UT [11].

a) Based on the process model Eq. (5), transform the sigma points.
b) The predicted state estimate at the time $(\tau + 1)$ with τ measurements is calculated as in Eq. (12)

$$S(\tau+1) = \sum_{i=0}^{2L_1} W_i^{(m)} S(i,(\tau+1)) \tag{12}$$

c) The predicted error covariance matrix, assuming additive and independent process noise, is calculated as in Eq. (13)

$$P(\tau+1) = \sum_{i=0}^{2L_1} W_i^{(c)} [S(i,(\tau+1)) - S_s(\tau+1)] \times [S(i,(\tau+1)) - S_s(\tau+1)]^T + Q(\tau) \tag{13}$$

d) The sigma points are updated using the predicted mean and predicted covariance as follows in the Eq. (14)

$$S(\tau+1) = \begin{bmatrix} S_s(\tau+1) \\ S_s(\tau+1) + \sqrt{(L_1+\lambda)+P(\tau+1)} \\ S_s(\tau+1) - \sqrt{(L_1+\lambda)+P(\tau+1)} \end{bmatrix}^T \tag{14}$$

e) Based on the measurement model given in Eq. (12), transform the predicted sigma points.

f) Predicted measurement matrix is calculated as in Eq. (15)

$$\hat{M}(\tau+1)=\sum_{i=0}^{2L_1}W_i^{(m)}Y(\tau+1) \tag{15}$$

where Y is given as in Eq. (16)

$$Y(\tau+1)=h(S(\tau+1)) \tag{16}$$

(a) The innovation covariance matrix is calculated as in Eq. (17)

$$P_{yy}=\sum_{i=0}^{2L_1}W_i^{(c)}\left[Y(i,(\tau+1))-\hat{M}(\tau+1)\right]\times\left[Y(i,(\tau+1))-\hat{M}(\tau+1)\right]^T+\sigma_B^2(\tau) \tag{17}$$

(b) The cross-covariance matrix is calculated as in Eq. (18)

$$P_{xy}=\sum_{i=0}^{2L_1}W_i^{(c)}[S(i,(\tau+1))-S_s(\tau+1)]\left[Y(i,(\tau+1))-\hat{M}(\tau+1)\right]^T \tag{18}$$

Kalman gain is calculated as in Eq. (19)

$$G(\tau+1)=P_{xy}P_{yy}^{-1} \tag{19}$$

(a) The estimated state is calculated as in Eq. (20)

$$S_S(\tau+1)=S(\tau+1)+G(\tau+1)\left(M(\tau+1)-\hat{M}(\tau+1)\right) \tag{20}$$

where $M(\tau+1)$is a matrix of measurement vector.

(a) The estimated error covariance matrix is updated as, given in Eq. (21)

$$P(\tau+1)=P(\tau+1)-G(\tau+1)P_{yy}G^T(\tau+1) \tag{21}$$

2.3. EKF Algorithm

The EKF linearizes the non-linearities in the state and measurement equations and then performs the Kalman filtering. Here the non-linearity is considered in the measurements obtained [24]. So, the measurement model matrix is linearized using Taylor series expansion and obtained as follows in Eq. (22)

$$H(\tau)=\begin{bmatrix}0 & 0 & \cos\beta(\tau)/R & -\sin\beta(\tau)/R\end{bmatrix} \tag{22}$$

where R is the range of the target from the observer obtained using Eq. (23)

$$R=\sqrt{(r_x(\tau))^2+(r_y(\tau))^2} \tag{23}$$

The covariance of the noise in the measurement equation is given as $\emptyset(\tau)$ which is a maximum level of Gaussian noise in bearings with a standard deviation of 0.33^0. The state vector time update Eq.(24) is given as

$$S_s(\tau+1)=A(\tau-1)*S_s(\tau) \tag{24}$$

The estimated state covariance matrix update Eq. (25) is given as

$$P(\tau+1,\tau)=A(\tau+1,\tau)*P(\tau-1)*(A(\tau,\tau))^T+Q(\tau+1) \tag{25}$$

The Kalman gain [2] for the EKF is given as Eq. (26)

$$G(\tau)=P(\tau+1,\tau)H^T(\tau)\left(H(\tau)*P(\tau+1,\tau)H^T(\tau)+\emptyset(\tau)\right)^{-1} \tag{26}$$

The measurement updates of the estimated state and estimated error covariance matrices are given respectively as follows in Eq. (27) and Eq. (28)

$$S_s(\tau+1,\tau+1)=S_s(\tau+1,\tau)+G(\tau)*Z(\tau) \tag{27}$$

$$P(\tau+1,\tau+1)=(I-G(\tau)*H(\tau))*P(\tau+1,\tau)*(I-G(\tau)*H(\tau))^T+G(\tau)*\emptyset(\tau)*(G(\tau))^T \tag{28}$$

The flowchart representing the EKF algorithm is as shown in Fig. 2. For each time sample, the estimated target parameters are evaluated against theoretically calculated true target parameters. If the acceptance criteria defined in Section 3 are achieved, then the solution is said to be obtained and the estimated parameters are forwarded to the weapon guidance cell for further process.

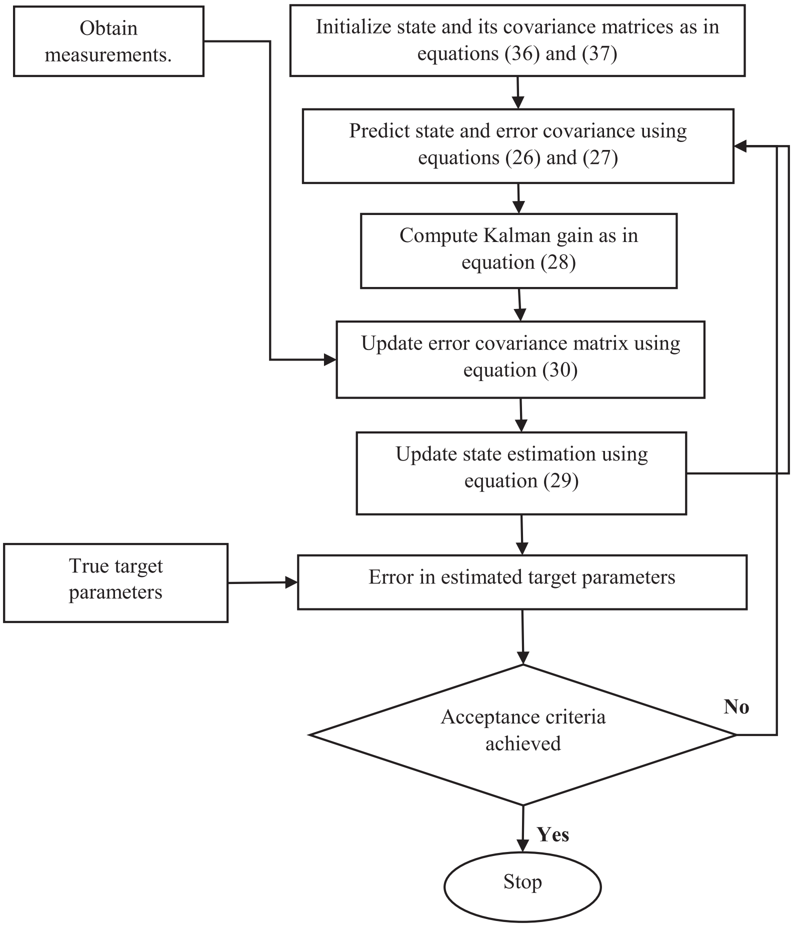

Obtain measurements.
Initialize state and its covariance matrices as in equations (36) and (37)
Predict state and error covariance using equations (26) and (27)
Compute Kalman gain as in equation (28)
Update error covariance matrix using equation (30)
Update state estimation using equation (29)
True target parameters
Error in estimated target parameters
Acceptance criteria achieved
No
Yes
Stop

Fig. 2. Flow chart representing EKF algorithm

2.4. Non-Gaussian noises

In this work, two forms of non-Gaussian noise conditions, SN and GMN, are considered meticulously. SN is generally due to statistical fluctuations in the signal itself and the statistical interaction process. The noise amplitude increases or decreases suddenly in the case of SN. Hence the same is designed in MATLAB using the "*randn*" function where the Gaussian noise is added with some random high amplitude noise as follows in Eq. (29)

$$Shot\ noise = \ Gauss\ noise\ +\ temp\ +\ 0.5 * randn \tag{29}$$

Here, *temp*is a random high-amplitude noise added to the Gaussian noise and (0.5**randn*)is added to make the variance of the distribution a random value. So, the SNcalculated will have a high amplitude and random variance.

The Gaussian mixture density applied has different variances varied from sample to sample and for which the data conditioned on the variance are normal. So, the GMN is generated by summing up of Gaussian noise with Gaussian noises of varying variances as given below.

$$Gaussmix = Gaussnoise + 2 * randn + randn + 4 * randn + 6 * randn \tag{30}$$

In Eq. (30), (2**randn*), (4**randn*), (6**randn*) individually generate Gaussian noises with different variances. These noises are added to form a GMNof random variance. Fig. 3 gives the histogram of all the three noise types simulated. These histograms give the count of different noise amplitudes used in the simulation of the algorithms with different noise types.

Fig. 3 gives the noise samples distribution during the simulation of 1800 samples. Fig. 3 (a) shows the distribution of Gaussian noise samples, where all the noise amplitudes are distributed between -1^{o} to 1^{o}. The noise samples are distributed according to the Gaussian distribution making the mean of noise amplitudes zero. Fig. 3 (b) gives the distribution of SN samples. The SN is applied only for some random samples and the rest of the samples will have Gaussian noise amplitude. It can be observed from Fig. 3 (b) that the SN samples are the ones having an amplitude greater than 1 degree. For about 800 random samples, noise with higher amplitudes is applied and the mean of noise amplitudes is also not zero, which reciprocates the applied SN samples. Fig. 3 (c) gives the distribution of GMN samples. GMN is a mixture of Gaussian noises

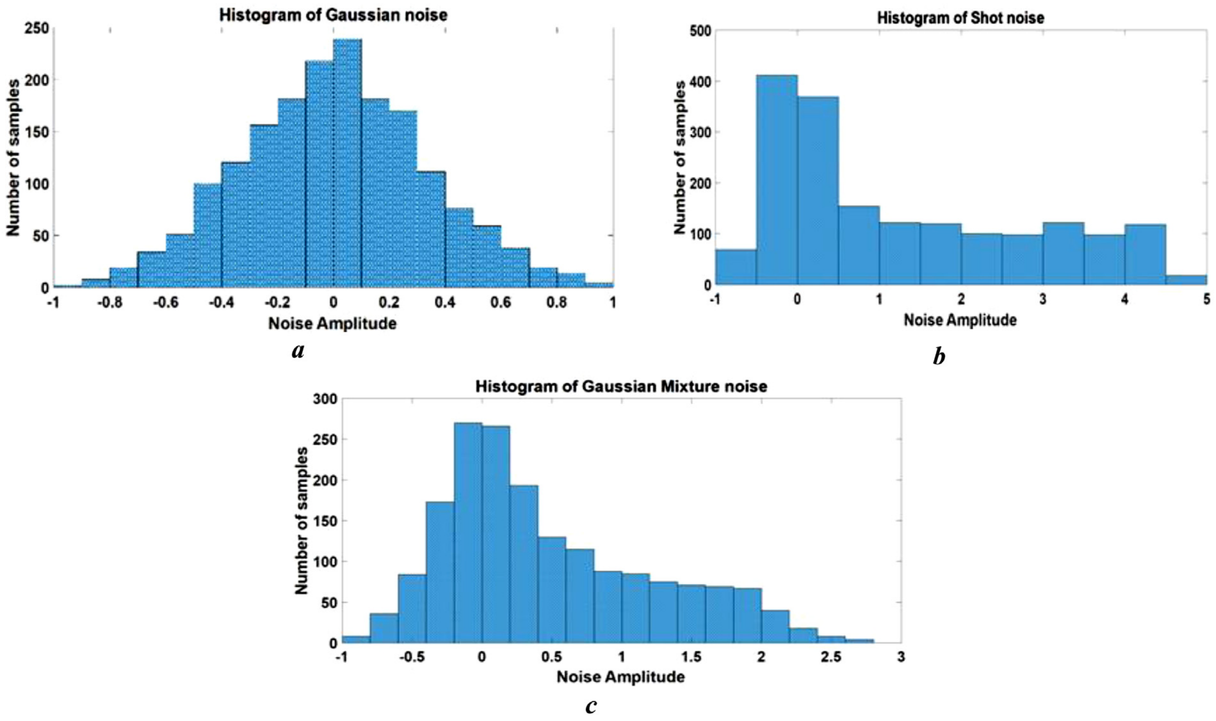


Fig. 3. Histogram of different noises used in this work. (a). Gaussian noise histogram. (b). Histogram of SN added to Gaussian noise. (c) Histogram of GMN added to Gaussian noise.

with different variances. So, it does not completely follow Gaussian distribution and is a combination of gaussian noises as shown in Fig. 3 (c).

2.5. Cramer-Rao lower bound

The Cramer-Rao lower bound (CRLB) provides a lower bound on the variance of an unbiased estimator for comparing the performance of any estimator. It is defined that for a nonlinear discrete-time system, the covariance of the estimated state follows the condition in Eq. (31)

$$E\left[\left(\hat{S}_S(\tau)-S_S(\tau)\right)\left(\hat{S}_S(\tau)-S_S(\tau)\right)^T\right]\geq J_\tau^{-1} \tag{31}$$

where J_τ is an information matrix, which is calculated recursively [9], using Eq. (32)

$$J_\tau = D_{\tau-1}^{22} - D_{\tau-1}^{21}\left(J_{\tau-1}+D^{11}\right)^{-1}D_{\tau-1}^{12} \tag{32}$$

where $D^{11}, D^{12}, D^{21} and\, D^{22}$, for additive Gaussian noise, are given as follows in Eq. (33)

$$D^{11} = A^TQ^{-1}A$$
$$D^{21} = -A^TQ^{-1}$$
$$D^{12} = \left(D^{21}\right)^T$$

$$D_{\tau-1}^{22} = Q^{-1} + E\left[H_\tau^T\left(\sigma_B^2\right)^{-1}H_\tau\right] \tag{33}$$

Here H_τ is given by the Jacobian of the measurement function,$H_\tau = [\;0 \quad 0 \quad \cos\hat{\beta}/R \quad -\sin\hat{\beta}/R\;]$ and $\hat{\beta}$is the predicted bearing measurement. The CRLB of range error iscalculated as in Eq. (34),

$$CRLB\_R_\tau = \sqrt{J_\tau^{-1}(3,3)+J_\tau^{-1}(4,4)} \tag{34}$$

Similarly, the CRLB of speed error is defined as in Eq. (35),

$$CRLB\_S_\tau = \sqrt{J_\tau^{-1}(1,1)+J_\tau^{-1}(2,2)} \tag{35}$$

CRLB of course error is calculated as in Eq. (36),

$$CRLB\_C_\tau = \tan^{-1}\left(\sqrt{J_\tau^{-1}(1,1)}/\sqrt{J_\tau^{-1}(2,2)}\right) \tag{36}$$

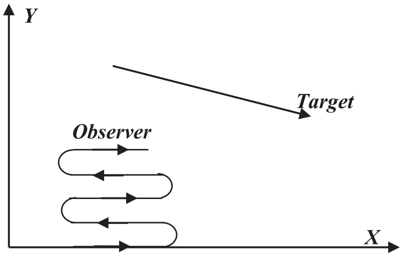


Fig. 4. Target and observer scenario

Table. 1
Scenarios considered for performance evaluation.

Scenario No.	Starting range (m)	Initial bearing (deg)	Speed of target (m/s)	Speed of observer (m/s)	The course of the target (deg)	ATB
1	3000	0	12	8	170	Low
2	3000	0	12	8	155	
3	3000	0	12	8	163	
4	3000	0	12	8	145	Medium
5	3000	0	12	8	140	
6	3000	0	12	8	148	
7	3000	0	12	8	135	High
8	3000	0	12	8	110	
9	3000	0	12	8	100	

3. Simulation and results

3.1. Initializations

This methodological researchassesses the efficiency of both algorithms through their implementation in the MATLAB PC setting.The measurements are presumed to be incessantly available for each second. In its course, the observer maneuvers to achieve the observability of the target. Firstly, the observer has a 90° course for two minutes and then turns 180° to hit the first line in maneuver, with a turning pace of 0.5° per second and attains a 270° course. The observer is considered to take four minutes for a complete maneuver of 180° as shown in Fig. 4. The target is assumed to be having different introductory courses in different scenarios, which is given in Table 1. With Gaussian noise, the standard deviation in the error is assumed to be 0.33°, which is 1σ error according to Gaussian distribution and 3σ error will be 1°. The values considered assuming that the advanced sonar has less probability of error in measurements. With non-Gaussian noise, the error in measurements given by sonar is assumed to be a little high and not more than 3°. So, the standard deviation in error with non-Gaussian noise is 1°, which is considered as 1σ error.

The target state vector's introductory estimate for implementation of both algorithms is taken as in Eq. (37)

$$S_s(0,0) = \begin{bmatrix} 5 & 5 & 5000\sin\beta_m & 5000\cos\beta_m \end{bmatrix}^T \tag{37}$$

The prediction of velocity components of the target is difficult as only angle measurements are available. So, they're presumed as 5m/s each.The introductory position of the target is calculated based on the day's sonar range, supposed to be 5000m.The initial value of the state covariance matrix may be taken as a diagonal matrix if the initial state estimation is uniformly distributed and is given as in Eq. (38)

$$P(0,0) = diagonal \begin{bmatrix} 4v_x^2(0,0)/12 \\ 4v_y^2(0,0)/12 \\ 4r_x^2(0,0)/12 \\ 4r_y^2(0,0)/12 \end{bmatrix} \tag{38}$$

The simulation and filtering for 100 Monte-Carlo runs are performed for low, medium, and high ATB scenarios mentioned in Table 1 using MATLAB [6] for both EKF and UKF algorithms. The performance is assessed based on the Root-Mean-Squared (RMS) error of the target parameters and the solution is obtained based on the criteria of acceptance for 100 Monte-Carlo runs explained as follows.

The acceptance criterion for 100 Monte-Carlo runs:

Error in estimated target range parameter<= (8/3)% of the true range

Error in estimated target course parameter<=1°.

Error in estimated target speed parameter<=0.33m/s.

In general, EKF and UKF are said not to be compatible with non-Gaussian noises. But the noise will not be completely non-Gaussian. So, it is considered that non-Gaussian noise occurs only at certain samples, and analysis of the consistency of

Table. 2
Time of convergence in seconds for scenarios with Gaussian noise.

Scenario	EKF				UKF			
	Range	Course	Speed	Convergence time	Range	Course	Speed	Convergence time
1	378	319	523	523	499	381	540	540
2	272	300	329	329	410	357	441	441
3	1592	1578	1592	1592	456	373	492	492
4	Fail	Fail	1727	Fail	342	305	374	374
5	282	254	251	282	366	341	395	395
6	267	302	330	330	366	328	393	393
7	271	311	318	318	375	365	399	399
8	338	415	338	415	491	498	515	515
9	343	423	235	423	533	590	553	590

Table. 3
Time of convergence in seconds for scenarios with SN.

Scenario	EKF				UKF			
	Range	Course	Speed	Convergence time	Range	Course	Speed	Convergence time
1	402	314	606	606	516	398	545	545
2	256	255	283	283	396	377	424	424
3	1603	1603	1603	1603	468	392	507	507
4	1648	1624	1635	1648	345	332	371	371
5	298	259	259	298	366	372	387	387
6	270	268	283	283	363	356	383	383
7	648	644	648	648	374	389	392	392
8	418	438	445	445	509	556	534	556
9	404	438	402	438	544	600	560	600

the filters to handle the noise is carried out. Convergence time is the point where the errors in estimated target parameters are within the acceptance criteria. Once the solution is converged, a weapon can be fired onto the target. The solution in real-time needs to be obtained in a short time, otherwise, there is a risk of getting tracked by the target. So, taking the convergence time to be below 10 minutes, the algorithms are evaluated for maximum noise amplitude of 3°. For EKF, the solutions were not obtained for most of the scenarios, and the convergence time, for which solution was obtained, was more than 900 seconds which is not worth in real-time tracking problems. As EKF is unstable in nature and the results are unpredictable with increasing non-linearity or noise, its performance is evaluated only for certain scenarios whose solution can be obtained.

3.2. Results

Table 2 represents the convergence times of the solution obtained for both EKF and UKF algorithms for low, medium, and high ATB scenarios with Gaussian noise. The simulation is carried out for 1800 seconds. The Gaussian noise is applied throughout the simulation period and as the measurements are assumed to be obtained every second, the noise is applied for all 1800 samples. For scenario 1 of the low ATB scenario mentioned in Table 1, the convergence times of range, course, and speed are 378s, 319s, and 523s respectively for the EKF algorithm. The total convergence time is 523s for the same scenario for EKF. The convergence times for the same low ATB scenario are 499s, 381s, and 540s respectively for the UKF algorithm and the total convergence time is 540s. Similarly, the convergence times are obtained for another low, medium, and high ATB scenarios. It can be observed from the data in Table 2 that EKF is unstable even when the noise is Gaussian when compared to UKF for 100 Monte-Carlo runs.

On the originally perceived line of sight the observer has to execute S-maneuver. So, for consistency in scenarios, the original bearing is selected as 0°. Scenario with an initial bearing of 30° is similar to the scenario with an initial bearing of 0°turned by 30°.

For EKF and UKF, Table 3 displays the convergence times obtained for 100 random SN samples with a median noise amplitude of 3°.SN is assumed to be occurring randomly at 100 samples out of 1800 samples taken for simulation. The data in Table 2 compared to the data in Table 3 respectively depicts the unstable and unpredictable performance of EKF, while the solutions obtained by UKF are consistent. For scenario 4, the convergence time was not obtained for EKF with the Gaussian noise environment as in Table 2, but for the same scenario, the convergence times are obtained in SN and Gaussian mixture environments as in Table. 3 and Table. 4 respectively. The convergence times obtained are also not within 600 seconds which is again of no use in case of tracking the target in real and hence instability in the algorithm of EKF is clearly understood.

For EKF and UKF, Table 4 reflects the convergence times obtained for 800 random GMN samples with 2.2° noise amplitude.GMN is assumed to be occurring randomly at 800 samples out of 1800 samples considered for simulation. The

Table. 4
Time of convergence in seconds for scenarios with GMN.

Scenario	EKF				UKF			
	Range	Course	Speed	Convergence time	Range	Course	Speed	Convergence time
1	348	997	361	997	500	424	545	545
2	358	906	361	906	416	409	447	447
3	446	997	910	997	453	421	498	498
4	362	998	361	998	372	381	395	395
5	1528	1572	1570	1572	465	482	573	573
6	358	997	358	997	385	391	401	401
7	286	301	907	907	387	424	401	424
8	489	996	817	996	501	583	516	583
9	Fail	1791	Fail	Fail	546	651	557	651

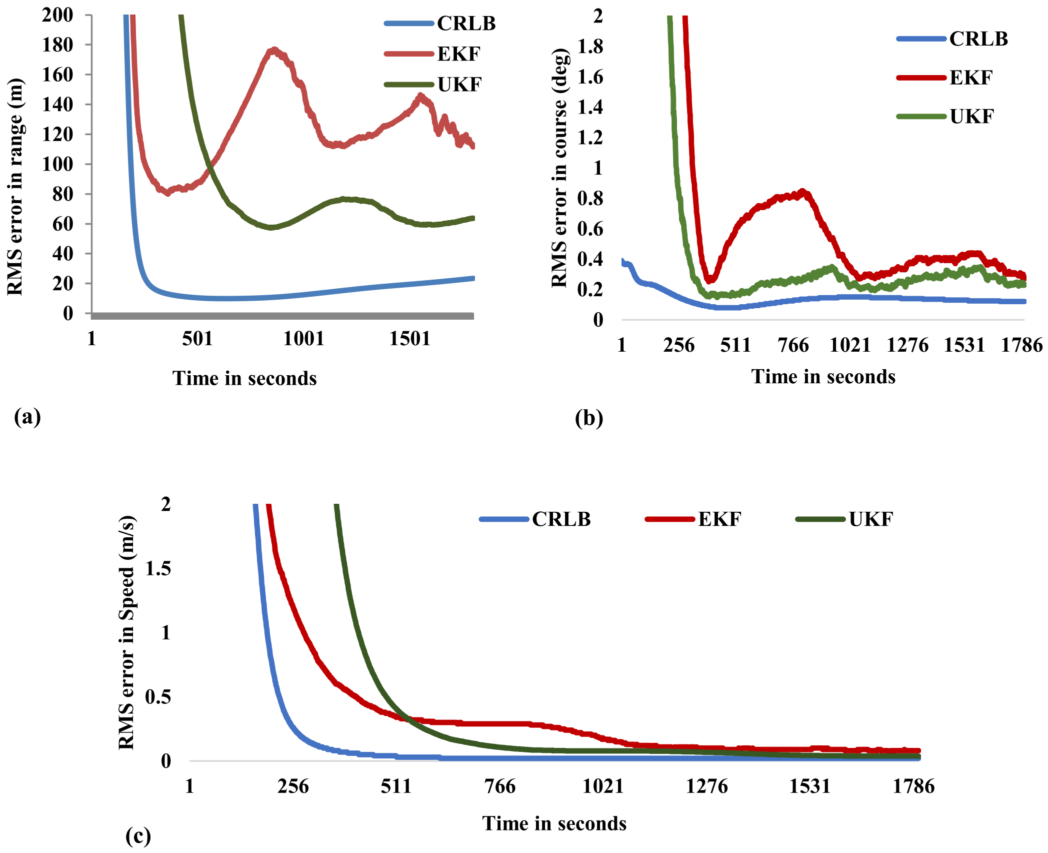


Fig. 5. (a) RMS error in estimated range with Gaussian Noise for Scenario 1 (b) RMS error in estimated course with Gaussian Noise for Scenario 1 (c) RMS error in estimated speed with Gaussian Noise for Scenario 1

amplitude of noise in measurement is taken care of not to exceed 2.2°. From the data in Table. 4, it can be inferred that EKF exhibits unstable nature while UKF provides compatible results.

In Fig. 5, Fig. 6 and Fig. 7, for low ATB scenario 1, the RMS errors in the target range, course, and speed calculations are shown for both EKF and UKF algorithms assuming that the noise is Gaussian noise, SN and GMN respectively. The RMS error in range, course, and speed estimates of the target is also compared with the theoretical CRLB to access the bias in estimators. It can be observed from the figures that UKF approaches the theoretical CRLB whereas the EKF estimator is far away from the bounds.

When SN was applied to algorithms periodically, the solution was obtained with a maximum of 72 samples for EKF and 180 samples for UKF. Similarly, when GMN was applied, the solutions were acquired with a maximum of 60 noise samples for EKF and 95 samples for UKF. The noise can't be periodic all the time. So, the non-Gaussian noises were applied to the algorithms at randomly selected samples of data. When SN was applied to randomly selected samples of data, the solution was obtained with a maximum of 90 samples for EKF and 200 samples for UKF. Similarly, when GMN was applied, the solutions were acquired with a maximum of 100 samples for EKF and 170 samples for UKF. However, with Gaussian noise,

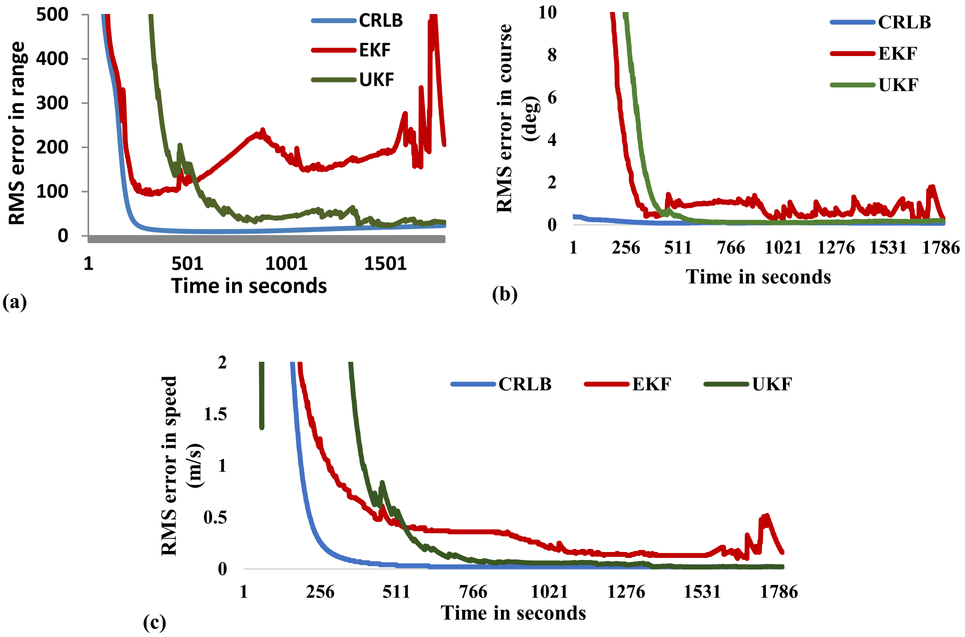


Fig. 6. (a) RMS error in estimated range with SN for Scenario 1 (b) RMS error in estimated course with SN for Scenario 1 (c) RMS error in estimated speed with SN for Scenario 1

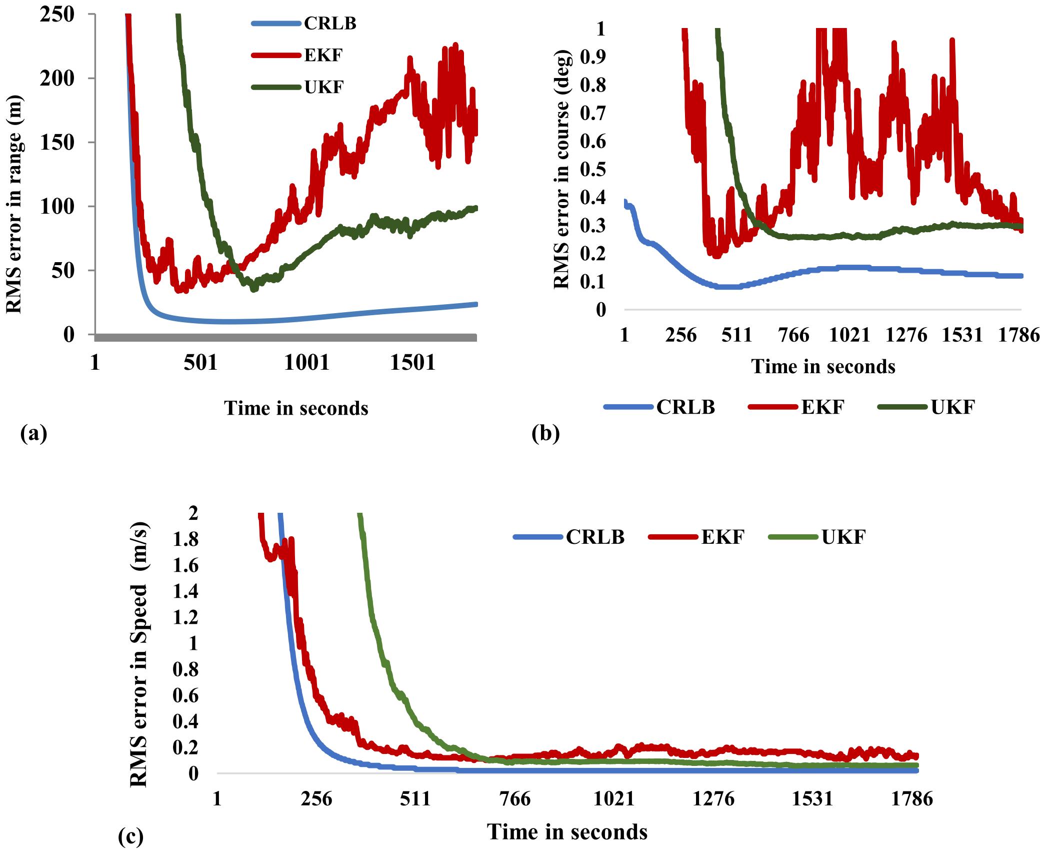


Fig. 7. (a) RMS error in estimated range with GMN for Scenario 1 (b) RMS error in estimated course with GMN for Scenario 1 (c) RMS error in estimated speed with GMN for Scenario 1

Table. 5
The percentage of non-Gaussian noise measurements that UKF can tolerate.

Noise type	Samples occurrence	Low ATB scenario	Medium ATB scenario	High ATB scenario
Shot noise	Continuous	33.3	33.3	3.5
Shot noise	Random	47.2	27.7	4.4
Gaussian mixture noise	Continuous	50	50	4.7
Gaussian mixture noise	Random	61	38.8	5.5

the algorithms work effectively even when the noise prevailed throughout the process. But, due to the unstable nature of EKF, the algorithm could not provide a solution to certain scenarios even in the presence of Gaussian noise.

3.3. Discussions

The analysis for UKF under the influence of the non-Gaussian noises is given in Table 5, with a maximum noise amplitude of 3°. The percentages of samples that can be incorporated with non-Gaussian noise for a maximum of 3° error and for which convergence time is below 600sec are only considered.

The noise samples are considered to occur in two different ways. One way of occurring noise samples is continuous, in which a non-Gaussian noise sample is assumed to be occurring at every 3rd sample or 4th sample, and so on. Another way of occurring noise samples is random, in which a non-Gaussian noise sample is assumed to be occurring randomly. The percentage of non-Gaussian noise samples is taken as the proportion of non-Gaussian noise samples to the total simulated samples. If the non-Gaussian noise is assumed to occur at every 3rd second, then the total non-Gaussian samples considered during simulation will be 600 and the percentage of no-Gaussian samples will be 33.3.

It can be concluded from Table 5 that UKF could take up to 47.2 percent, 27.7 percent, and 4.4 percent of measurements with SN for low, medium, and high ATB scenarios respectively, when the noise observations are known to arise spontaneously.Similarly, for low, medium, and high ATB scenarios, it can survive up to 61 percent, 38.8 percent, and 5.5 percent of random GMN samples.So, from the simulation, it is evident that UKF has better stability than EKF.From the simulation findings, it was found that UKF operates well with both forms of noise whereas EKF just functions within certain limits.When noise continues to increase, the EKF algorithm becomes extremely unstable.It is therefore difficult to establish the percentage of samples EKF may sustain. As the ATB increases, the target moves away from the observer, hence the percentage of samples that the UKF algorithm can tolerate decreases as the range increases. Nevertheless, the UKF shows consistent results for fewer noise ranges than the EKF.

4. Conclusion

In the investigaton to evaluate the filters with two separate non-Gaussian noise conditions namely SN and GMN. The results show the efficiency of UKF and EKF in providing accurate solutions when different kinds of noises are present.(clarity of performance of UKF and EKF missing). It was found, from the simulation results, that UKF performs well with both forms of noise for low and medium ATB scenarios while with high ATB scenarios the efficacy is poor as the range increases.The findings show that UKF will take up to 33 percentfor low ATB scenarios, 27 percent for medium ATB scenarios, and 3.5 percent of samples for high ATB scenarios with SN. Similarly, with GMN samples,it can tolerate up to 50 percent, 38 percent, and 4 percent of samples for low, medium, and high ATB situations. On the other side, EKF rarely operates within certain limits and is highly unpredictable. Even for a lesser percentage of non-Gaussian samples, the solution was not obtained with EKF for many scenarios. Nonetheless, UKF shows consistent results with smaller noise tests than EKF.

Declaration of Competing Interest

The authors declare that they have no known competing financialinterestsor personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research is supported by the Department of Science and Technology (DST) under the WOS-A scheme via a sponsored project: SR/WOS-A/ET-139/2017(G). The authors would like to acknowledge DST, New Delhi, and President, Koneru Lakshmaiah Education Foundation (Deemed to be University) for their continuous support and encouragement.

References

[1] Aidala VJ. Kalman filter behavior in bearings-only tracking applications. IEEE Trans Aerosp Electron Syst 1979;AES-15(1):29–39.
[2] Nardone SC, Lindgren AG, Gong KF. Fundamental properties and performance of conventional bearings-only target motion analysis. IEEE Trans Autom Control September 1984;AC-29(9):775–87.
[3] Nardone SC, Aidala VJ. Observability criteria for bearings-only target motion analysis. IEEE Trans Aerosp Electron Syst 1981;AES-17(2):162–6.

[4] Koteswara Rao S. Comments on Discrete-time observability and estimability analysis for bearings-only target motion analysis. IEEE Trans Aerosp Electronic Syst 1998;32(4):1361–7.

[5] Weiliang Zhu; Zhaopeng Xu; Bo Li; Zhidong Wu, "Research on the observability of bearings-only target tracking based on multiple sonar sensors", second international conference on instrumentation, measurement, computer, communication, and control; pp: 631-634, 10.1109/IMCCC.2012.154; IEEE Conference Publications, 2012.

[6] Wan EA, Van Der Merwe Rudolph. "The unscented Kalman filter for nonlinear estimation". In: Proc. IEEE Symposium 2000 on adaptive systems for signal processing, communication, and control. Alberta, Canada; October 2000. p. 153–8.

[7] Julier SJ, Uhlmann JK. Unscented filtering and nonlinear estimation. Proc. IEEE 2004;92(3):401–22.

[8] Koteswara Rao S, Raja Rajeswari K, Linga Murthy KS. Unscented Kalman filter with application to bearings-only target tracking. IETE J. Res. March-April. 2009;55(2):63–7.

[9] Ristick, B., Arulampalam, M. S., and Gordon, N., "Beyond Kalman filters–particle filters for tracking applications", Artech House, DSTO, 2004.

[10] Brehard T, Cadre Jean-pierre Le. Closed-form posterior Cramér-Rao Bound for a manoeuvring target in the bearings-only tracking context using best-fitting Gaussian distribution. 9th international conference on information fusion; February 2007. doi:10.1109/ICIF.2006.301625.

[11] Simon Dan. Optimal state estimation: Kalman. H∞ and nonlinear Approximations. Wiley; 2006.

[12] Karlsson R, Gustafsson F. Recursive Bayesian estimation: bearings-only applications. IEE Proc. Radar, Sonar & Navig. October 2005;152(5):305–13.

[13] Clark Martin, Maskell Simon, Yaqoob Richard VinterandMoeen. A comparison of the particle and shifted Rayleigh filters in their application to a multi-sensor bearings-only problem. In: IEEE aerospace conference; 2005. p. 2142–7.

[14] Radhakrishnan R, Bahumik S, Singh AK, Tomar NK. Quadrature filters for underwater passive bearings-only target tracking. Sens. Signal Proces. Def. (SSPD), Edinburgh September 2015.

[15] Sharma Suresh Kumar, J Manisha. Nene, "Performance evaluation of unscented Kalman Filter using multi-core processors environment. International conference on computer, communication and control (IC4); September 2015.

[16] Liu Jingfen, Wang Yuchen, Wang Zheng. A novel hybrid estimator for real-time bearings-only target tracking", Chinese Control and Decision Conference (CCDC). In: IEEE conference publications; 2016. p. 3900–5.

[17] Cappe O, Godsill SJ, Moulines E. An overview of existing methods and recent advances in sequential Monte Carlo. Proc. IEEE 2007;95(5):899–924.

[18] Alspach DL, Sorenson HW. Nonlinear Bayesian estimation using Gaussian sum approximations. IEEE Trans. Autom. Control 1972;17(4):439–48.

[19] Terejanu G, Singla P, Singh T, Scott PD. Adaptive Gaussian sum filter for nonlinear Bayesian estimation. IEEE Trans. Autom. Control 2011;56(9):2151–6.

[20] Fu XY, Jia YM. An improvement on resampling algorithm of particle filters. IEEE Trans. Signal Proces. 2010;58(10):5414–20.

[21] Kabaoglu N. Target tracking using particle filters with support vector regression. IEEE Trans. Veh. Technol. 2009;58(5):2569–73.

[22] Leong PH, Arulampalam S, Lamahewa TA, Abhayapala TD. A Gaussian-sum based cubature Kalman filter for bearings-only tracking. IEEE Trans. Aerosp. Electron. Syst. 2013;49(2):1161–76.

[23] Terejanu G. An adaptive split-merge scheme for Uncertainty propagation using Gaussian mixture models. proceedings of the 49th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition. Orlando, Fla, USA; January 2011.

[24] Jahan Kausar, Koteswara Rao S. Extended Kalman Filter for bearings-only tracking. Int. J. Eng. Adv. Technol. (IJEAT), Vol. 8, No. 6, pp 637–640, Aug' 2019.

Kausar Jahan, a research scholar in Koneru Lakshmaiah Educational Foundation, Vaddeswaram, Guntur, AP. India. She completed her B. Tech in 2009 at JNTU Kakinada and M Tech in 2015 at JNTU Kakinada in ECE. She is presently working as a women scientist WOS-A.

S. Koteswara Rao, former scientist "G" in NSTL, DRDO, Visakhapatnam is currently working as Professor in Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, AP, India. He published several papers in International Conference & Journals in the field of signal processing. He is the fellow member of IETE.