Article

Fusion of Angle Measurements from Hull Mounted and Towed Array Sensors

Kausar Jahan * and **Koteswara Rao Sanagapallea**

Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur 522502, Andhra Pradesh, India; skrao@kluniversity.in
* Correspondence: kausar465@ieee.org

Abstract: Two sensor arrays, hull-mounted array, and towed array sensors are considered for bearings-only tracking. An algorithm is designed to combine the information obtained as bearing (angle) measurements from both sensor arrays to give a better solution. Using data from two different sensor arrays reduces the problem of observability and the observer need not follow the S-maneuver to attain observability of the process. The performance of the fusion algorithm is comparable to that of theoretical Cramer–Rao lower bound and with that of the algorithm when bearing measurements from a single sensor array are considered. Different filters are used for analyzing both algorithms. Monte Carlo runs need to be done to evaluate the performance of algorithms more accurately. Also, the performance of the fusion algorithm is evaluated in terms of solution convergence time.

1. Introduction

The demand for fast and accurate tracking of the targets is increasing day by day. Under water, noise corrupted angle measurements are available from noise radiated from the target and processed from a single observer platform operating in passive mode [1]. Passive target tracking is more popular as it helps the tracker from being tracked. In the ocean environment, target motion analysis is generally performed in a two-dimensional plane using angles-only measurements. The angles generated by the sensors are well known as bearing angles. The observer with the single sensor array, say hull-mounted array (HMA), in a passive mode, determines parameters of target motion like its range, course, bearing angle, speed, etc. As the range of the target is not available and bearing being nonlinearly related to target motion parameters, nonlinearity in the process is high [2]. In addition to this, measurements from a single sensor array are only considered to be available, which makes the process unobservable and the observer must perform a proper maneuver to make the process observable [3–5].

The ocean environment is unstable all the time and this makes the bearing measurements from a single sensor array unreliable in obtaining all the required information about the target. Moreover, an increase in system complexity increases the need for more input from different sensor arrays to obtain complete information on the system. According to the application and need, a system with multiple sensors can choose different sensors with the required characteristics and based on the environmental conditions for obtaining the target information. Information from these sensors is then consistently combined coherently to obtain the parameters like range, course, velocity, and acceleration of the target during its movement. This is known as data fusion problem. As the bearing measurements are taken from two sensors, in this work, the maneuver of the observer to obtain the observability of the target is reduced in most of the scenarios.

Angles-only tracking, commonly known as bearings-only tracking, used classical filters like maximum likelihood estimator, pseudo linear estimator, etc. in the early 19th century [1,6] and these were batch processing filters. In real-time scenarios, these batch processing estimators need a lot of time for obtaining the estimated results. This led to the evolution of recursive estimators like least square estimator, Kalman filter (KF), etc. KF being a linear filter was not suitable for nonlinear bearings-only tracking [3]. The nonlinearity in the process leads to the evolution of extended KF (EKF), unscented KF (UKF), particle filter (PF), etc. to deal with the nonlinearities in the process and measurements [7]. EKF reduces the nonlinearities by linearizing the measurement equation using Taylor series expansion and updates the state and covariance matrices and calculates the filter gain. The filter is mostly dependent on the initializations and due to the linearization process; information is lost making the solution unreliable. In UKF, the probability density function of the state vector is propagated through a set of carefully chosen sample points, called sigma points, using unscented transform [8,9]. This helps in possession of posterior mean and covariance up to second order for any type of nonlinearity. Hence, unlike EKF, UKF can provide improved stability, accuracy, and better convergence in solution [10–12].

Y. Bar-Shalom, in [13], gave a deep insight regarding data association and data fusion of multiple sensors in multiple target environment using radar measurements. Several filtering algorithms were utilized including EKF, UKF, and PF. Data fusion for underwater surveillance was presented in [14–16], in which the technique was evaluated only using MGBEKF and UKF algorithms. In [17], the author used a data fusion technique to evaluate the performance of the fusion technique for two and more than two sensors of data fusion. It is an extensive work for the work presented by Bar-Shalom on data fusion. In [18], the authors applied a data fusion technique using a PF for estimation of robot motions. In [19], the authors analyzed the fusion technique for batch processing using the maximum likelihood estimator and analyzed the same with recursive processing with an unscented Gauss–Helmert Kalman filter. The observability of the system is further discussed in the paper. In [18], a data fusion algorithm was used along with a data association algorithm to detect multiple targets. It deals with the hardware complexity in target detection. In [19], data fusion was used to detect the target in a properly designed underwater network. In [20], authors used interacting multiple model techniques combined with EKF and UKF for tracking the single as well as multiple targets and compared two different data fusion techniques that involve fusion using nearest neighborhood calculation and fusion using probabilistic data association. The data fusion techniques in literature are extended to robotics, in air surveillance, and underwater surveillance applications. However, there is a necessity to explore the fusion techniques in detail for underwater surveillance applications.

The filters considered for evaluation of fusion algorithm are further compared with theoretical Cramer–Rao lower bound (CRLB). The mathematical modelling of CRLB for tracking algorithms using a single sensor array is in [9,21] and that for tracking algorithms using two sensor arrays is represented in Section 2.6.

In this paper, a single target following the constant velocity model is considered for simplicity. The fusion of measurement information from two sensor arrays, HMA and towed array (TA) is considered without maneuver. TA is towed behind the observer at a certain distance away from the observer. This helps in the reduction of self-noise from the observer and allows it to operate at very low frequencies. Hence, the TA will be capable of even tracking the target at great distances. The algorithm is evaluated for several scenarios in Monte Carlo simulations.

In this paper, the performance of BOT with a single sensor (HMA) and two sensors (HMA and TA) using different filters is evaluated. The mathematical modelling of all the algorithms is described in Section 2. The results obtained from the simulation of the algorithms and discussion on the results is given in Section 3. This paper is concluded in Section 4.

2. Mathematical Modeling

2.1. Tracking Using a Single Sensor Array

Tracking using a single sensor array (SST) measurement considers the bearing measurements from HMA only. The observer is assumed to be following constant speed with S—the maneuver in the course and the target follows persistence speed and course [20]. At time instant, say 'τ', state vector of the observer [9,10] is given as Equation (1).

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

Here $v_{xo}(\tau)$, $v_{yo}(\tau)$, $r_{xo}(\tau)$, $r_{yo}(\tau)$ are the respective speeds and ranges of the observer in x and y coordinates. With the known speed and course of the observer, change in the observer's x and y position is obtained as Equations (2) and (3).

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

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

Here $dr_{xo}(\tau)$ is the change in x coordinate and $dr_{yo}(\tau)$ is the change in y coordinates of observer. ocr is the observer course with τ being the time of a second. The state vector of the target is given as Equation (4).

$$S_t(\tau) = \begin{bmatrix} v_{xt}(\tau) & v_{yt}(\tau) & r_{xt}(\tau) & r_{yt}(\tau) \end{bmatrix}^T. \tag{4}$$

Here $v_{xt}(\tau)$, $v_{yt}(\tau)$, $r_{xt}(\tau)$, $r_{yt}(\tau)$ represent target's speed and range values in x and y coordinates respectively. Target position, when its course and speed are known, changes according to the following Equations (5) and (6).

$$dr_{xt}(\tau) = v_{xt}(\tau)\sin(tcr)\tau, \tag{5}$$

$$dr_{yt}(\tau) = v_{yt}(\tau)\cos(tcr)\tau. \tag{6}$$

Here $dr_{xt}(\tau)$, $dr_{yt}(\tau)$ are the progress in x and y coordinates of the target. tcr is the target course and τ is the time of one second. The target's state vector relative to the observer [9,12,13] is as in Equation (7).

$$S_s(\tau) = \begin{bmatrix} v_x(\tau) & v_y(\tau) & r_x(\tau) & r_y(\tau) \end{bmatrix}^T. \tag{7}$$

Here $v_x(\tau)$, $v_y(\tau)$, $r_x(\tau)$, $r_y(\tau)$ are relative values of respective speed and range in x and y coordinates. The state vector of the target relative to the observer for the proceeding time, concerning the relative target state vector at present is given as Equation (8).

$$S_s(\tau+1) = A(\tau)S_s(\tau) + b(\tau+1) + \omega C(\tau). \tag{8}$$

Here $A(\tau)$ is the system dynamics matrix given as Equation (9) and $b(\tau)$ is the deterministic matrix given as in Equation (10).

$$A(\tau) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \tau & 0 & 1 & 0 \\ 0 & \tau & 0 & 1 \end{bmatrix}, \tag{9}$$

$$b(\tau+1) = \begin{bmatrix} 0 & 0 & -(r_{xo}(\tau+1) - r_{xo}(\tau)) & -\left(r_{yo}(\tau+1) - r_{yo}(\tau)\right) \end{bmatrix}^T. \tag{10}$$

Here $C(\tau)$ is the system noise and its gain matrix, ω is taken as in Equation (11).

$$\omega = \begin{bmatrix} \tau & 0 \\ 0 & \tau \\ \tau^2/2 & 0 \\ 0 & \tau^2/2 \end{bmatrix} \tag{11}$$

Let $Q(\tau)$ be the covariance of system noise. Then it is given as in Equations (12) and (13).

$$Q(\tau) = E\big[(\omega C(\tau))(\omega C(\tau))^T\big], \tag{12}$$

$$Q(\tau) = \sigma^2 \begin{bmatrix} \tau^2 & 0 & \tau^3/2 & 0 \\ 0 & \tau^2 & 0 & \tau^3/2 \\ \tau^3/2 & 0 & \tau^4/4 & 0 \\ 0 & \tau^3/2 & 0 & \tau^4/4 \end{bmatrix}. \tag{13}$$

Here σ^2 is taken as the variance of system noise multiplied with some constant value as given in (13). The measurement equation for SST has bearing angles from single sensor array (HMA) and the bearing angle $\beta(\tau)$ is given as in Equation (14).

$$\beta(\tau) = \tan^{-1}\big(r_x(\tau)/r_y(\tau)\big). \tag{14}$$

Measured bearing, $\beta_m(\tau)$, degraded with noise is obtained and is mathematically expressed as in Equation (15).

$$\beta_m(\tau) = \beta(\tau) + \eta(\tau). \tag{15}$$

Here $\eta(\tau)$ is the noise in the measurement. Here zero-mean additive process noise and measurement noise are considered. The system measurement equation is given as in Equation (16).

$$Z(\tau) = H(\tau)S_s(\tau) + \gamma(\tau). \tag{16}$$

Here $H(\tau)$ matrix gives the relation between measurement and state of the system and the measurement noise matrix is represented by $\gamma(\tau)$.

2.2. Tracking Using Two Sensor Arrays

For target tracking using two sensor arrays (TST), the bearing measurements from HMA and TA are considered. Consider the true bearing angles from HMA and TA be $\beta_{ha}(1)$, $\beta_{ha}(2)$, $\beta_{ha}(3)$, …, $\beta_{ha}(\tau)$ and $\beta_{ta}(1)$, $\beta_{ta}(2)$, $\beta_{ta}(3)$, …, $\beta_{ta}(\tau)$ respectively where τ represents the time. Measurement vector, $\beta_m(\tau)$ can be written as in Equation (17).

$$\beta_m(\tau) = \begin{bmatrix} \beta_{ham}(\tau) & \beta_{tam}(\tau) \end{bmatrix}^T, \tag{17}$$

τ represents the time for which the measurement vector is considered. The relative target state vector, that is taken concerning observer be $S_S(\tau)$, is given as in Equation (18).

$$S_S(\tau) = \begin{bmatrix} \dot{v}_x(\tau) & \dot{v}_y(\tau) & v_x(\tau) & v_y(\tau) & r_x(\tau) & r_y(\tau) \end{bmatrix}^T. \tag{18}$$

Here, $\dot{v}_x(\tau)$, $\dot{v}_y(\tau)$, $v_x(\tau)$, $v_y(\tau)$, $r_x(\tau)$, and $r_y(\tau)$ are corresponding acceleration, corresponding speed and corresponding range components of the target concerning observer. The dynamic equation of the target state is given as in Equations (19) and (20).

$$S_S(\tau+1) = A(\tau+1)S_S(\tau) + b(\tau+1) + \omega C(\tau), \tag{19}$$

$$b(\tau+1) = \begin{bmatrix} 0 & 0 & 0 & 0 & -(r_{xo}(\tau+1) - r_{xo}(\tau)) & -(r_{yo}(\tau+1) - r_{yo}(\tau)) \end{bmatrix}^{T}. \tag{20}$$

Here A is the transition matrix and is given as in Equation (21) and ω is taken as in Equation (22).

$$A(\tau+1) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \tau & 0 & 1 & 0 & 0 & 0 \\ 0 & \tau & 0 & 1 & 0 & 0 \\ \tau^2/2 & 0 & \tau & 0 & 1 & 0 \\ 0 & \tau^2/2 & 0 & \tau & 0 & 1 \end{bmatrix}, \tag{21}$$

$$\omega = \begin{bmatrix} 1 & 0 & \tau & 0 & \tau^2/2 & 0 \\ 0 & 1 & 0 & \tau & 0 & \tau^2/2 \end{bmatrix}^{T}. \tag{22}$$

The plant noise $\omega(\tau)$ is assumed to follow Gaussian distribution. It is given by Equation (23).

$$C(\tau) = \begin{bmatrix} C_{\ddot{x}} & C_{\ddot{y}} \end{bmatrix}^{T}. \tag{23}$$

The plant noise covariance matrix is given Equations (24) and (25).

$$Q(\tau) = E\left[\omega C(\tau)C^{T}(\tau)\omega^{T}\right], \tag{24}$$

$$Q(\tau) = \begin{bmatrix} 1 & 0 & \tau & 0 & \tau^2/2 & 0 \\ 0 & 1 & 0 & \tau & 0 & \tau^2/2 \\ \tau & 0 & \tau^2 & 0 & \tau^3/2 & 0 \\ 0 & \tau & 0 & \tau^2 & 0 & \tau^3/2 \\ \tau^2/2 & 0 & \tau^3/2 & 0 & \tau^4/4 & 0 \\ 0 & \tau^2/2 & 0 & \tau^3/2 & 0 & \tau^4/4 \end{bmatrix} \times E\left[C(\tau)C^{T}(\tau)\right]. \tag{25}$$

Initially, HMA is assumed to be at origin and TA is L meters apart from HMA and towed behind the ship. Figure 1 represents the initial positions of HMA, TA, and target. Let (R_{x1}, R_{y1}) and (R_{x2}, R_{y2}) be x and y coordinates of target range corresponding to HMA and TA. The observer is assumed to be moving with a course of α concerning the Y-axis. B_{ta} and B_{ha} are the true bearing angles concerning TA and HMA. (R_{x3}, R_{y3}) is the location of TA concerning HMA and is assumed to be 500m apart from HMA.
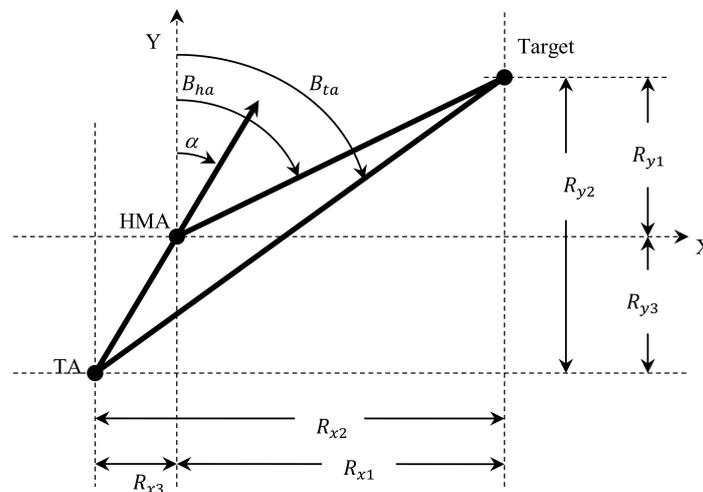
Figure 1. The observer with two arrays.

From Figure 1, R_{x3} and R_{y3} are given by Equations (26) and (27).

$$R_{x3} = L\sin(\alpha + 180), \tag{26}$$

$$R_{y3} = L\cos(\alpha + 180). \tag{27}$$

From Figure 1, the relation between range coordinates can be written as in Equations (28) and (29).

$$R_{x2} = R_{x1} + R_{x3}, \tag{28}$$

$$R_{y2} = R_{y1} + R_{y3}. \tag{29}$$

The HMA bearing, β_{ham} is modelled as in Equation (30).

$$\beta_{ham}(\tau) = \tan^{-1}\left(\frac{R_{x1}(\tau)}{R_{y1}(\tau)}\right) + \gamma_{ha\beta}(\tau), \tag{30}$$

where $\gamma_{ha\beta}(\tau)$ is an error in the HMA measurement and is assumed to follow zero-mean Gaussian distribution with variance $\sigma^2_{ha\beta}(\tau)$. The measurement and plant noises are assumed to be uncorrelated to each other. Measured bearing from TA, $\beta_{tam}(\tau)$ is given as in Equation (31).

$$\beta_{tam}(\tau) = \tan^{-1}\left(R_{x2}(\tau)/R_{y2}(\tau)\right) + \gamma_{ta\beta}(\tau). \tag{31}$$

Here, error in TA measurement is represented as $\gamma_{ta\beta}(\tau)$ and is assumed to follow Gaussian distribution with zero mean and having variance of $\sigma^2_{ta\beta}(\tau)$.

The covariance matrix of the measurements, $\varnothing(\tau)$, at time τ is taken as in Equation (32).

$$\varnothing(\tau) = \begin{bmatrix} \sigma^2_{ha\beta}(\tau) & 0 \\ 0 & \sigma^2_{ta\beta}(\tau) \end{bmatrix}. \tag{32}$$

2.3. Algorithm of Extended Kalman Filter for TST

The main idea of EKF is to linearize the existing non-linearities in measurement as well as state equations and then perform Kalman filtering. The nonlinearity in this application is considered only in the measurement Equation (16) [20]. Therefore, a partial derivation is applied to the measurement model matrix for linearization and is linearized measurement model matrix for HMA, H_{ha} is obtained as follows in Equation (33).

$$H_{ha}(\tau+1) = \begin{bmatrix} 0 & 0 & 0 & 0 & \dfrac{\hat{R}_{y1}(\tau+1,\tau)}{\hat{R}_1^2(\tau+1,\tau)} & \dfrac{-\hat{R}_{x1}(\tau+1,\tau)}{\hat{R}_1^2(\tau+1,\tau)} \end{bmatrix}. \tag{33}$$

Here $R_1^2 = R_{x1}^2 + R_{y1}^2$

The true parameters of the target are not known in real-time scenarios, so the estimated values, $\hat{R}_{x1}$, and $\hat{R}_{y1}$ are used in Equation (33). Measurement matrix of TA, H_{ta} is given by Equation (34).

$$H_{ta}(\tau+1) = \begin{bmatrix} 0 & 0 & 0 & 0 & \dfrac{\hat{R}_{y2}(\tau+1,\tau)}{\hat{R}_2^2(\tau+1,\tau)} & \dfrac{-\hat{R}_{x2}(\tau+1,\tau)}{\hat{R}_2^2(\tau+1,\tau)} \end{bmatrix}. \tag{34}$$

Here $R_2^2 = R_{x2}^2 + R_{y2}^2$.

Measurement fused matrix, H is given by Equation (35).

$$H(\tau+1) = \begin{bmatrix} H_{ha}(\tau+1) & H_{ta}(\tau+1) \end{bmatrix}^T. \tag{35}$$

The priori state vector estimation is calculated as follows in Equation (36).

$$S_s(\tau + 1|\tau) = A(\tau)S_s(\tau|\tau). \tag{36}$$

The estimated priori state covariance matrix estimation [2] is done as follows in Equation (37).

$$P(\tau + 1|\tau) = A(\tau)P(\tau|\tau)(A(\tau))^T + Q(\tau). \tag{37}$$

The gain [2] is calculated for EKF as in Equation (38).

$$G(\tau) = P(\tau + 1|\tau)H^T(\tau)\big(H(\tau)P(\tau + 1|\tau)H^T(\tau) + \varnothing(\tau)\big)^{-1}. \tag{38}$$

The posteriori state and its covariance matrices, based on the measurements, are updated respectively as follows in Equations (39) and (40).

$$S_s(\tau + 1|\tau + 1) = S_s(\tau + 1|\tau) + G(\tau)Z(\tau), \tag{39}$$

$$P(\tau + 1|\tau + 1) = (I - G(\tau)H(\tau))P(\tau + 1|\tau)(I - G(\tau)H(\tau))^T + G(\tau)\varnothing(\tau)(G(\tau))^T. \tag{40}$$

2.4. Algorithm of Modified Gain Bearings-Only Extended Kalman Filter for TST

The modified gain function, g_{ha} [12] is given by Equation (41).

$$g_{ha} = \begin{bmatrix} 0 & 0 & 0 & 0 & \left(\dfrac{\cos\beta_{ham}}{\hat{R}_{x1}\sin\beta_{ham} + \hat{R}_{y1}\cos\beta_{ham}} \right) & \left(\dfrac{-\sin\beta_{ham}}{\hat{R}_{x1}\sin\beta_{ham} + \hat{R}_{y1}\cos\beta_{ham}} \right) \end{bmatrix}, \tag{41}$$

and TA's modified gain function, g_{ta} can be written as in Equation (42).

$$g_{ta} = \begin{bmatrix} 0 & 0 & 0 & 0 & \left(\dfrac{\cos\beta_{tam}}{\hat{R}_{x2}\sin\beta_{tam} + \hat{R}_{y2}\cos\beta_{tam}} \right) & \left(\dfrac{-\sin\beta_{tam}}{\hat{R}_{x2}\sin\beta_{tam} + \hat{R}_{y2}\cos\beta_{tam}} \right) \end{bmatrix}. \tag{42}$$

Finally, g is written as in Equation (43).

$$g = \begin{bmatrix} g_{ha} & g_{ta} \end{bmatrix}^T. \tag{43}$$

The predicted covariance matrix is calculated as in Equation (44).

$$P(\tau + 1) = \big(A(\tau + 1)P(\tau)A^T(\tau + 1)\big) + \omega C(\tau + 1)\omega^T. \tag{44}$$

The Kalman gain is calculated as in Equation (45).

$$G(\tau + 1) = P(\tau + 1)H^T(\tau + 1)\big[\varnothing(\tau) + H(\tau + 1)P(\tau + 1)H^T(\tau + 1)\big]^{-1}. \tag{45}$$

The updated state matrix is calculated as in Equation (46).

$$S_s(\tau + 1) = S_s(\tau + 1) + G(\tau + 1)[Z(\tau + 1) - Z(\tau + 1, S_s(\tau + 1))]. \tag{46}$$

Here $Z(\tau + 1, S_s(\tau + 1))$ is the bearing measurement obtained from predicted estimate at time index $(\tau + 1)$. The updated covariance matrix is given in Equation (47).

$$\begin{aligned} P(\tau + 1) = [I - G(\tau + 1)g(\beta_m(\tau + 1), S_s(\tau + 1))] \\ P(\tau + 1)[I - G(\tau + 1)g(\beta_m(\tau + 1), S_s(\tau + 1))]^T + \varnothing(\tau)G(\tau + 1)G^T(\tau + 1). \end{aligned} \tag{47}$$

2.5. Particle Filter Combined with Other Filters

PF [12] is an advanced nonlinear filter, suitable for systems that are nonlinear and applicable in the non-Gaussian environment. In PF, estimation is done by considering the probability density function (PDF) of the state vector. Initially, a set of N random state vectors are generated based on the initial PDF of the state, $p(S_S(0|0))$, which is assumed to be a known value [21]. These randomly generated state vectors are called particles and are represented as $S_S(n, \tau|\tau)(n = 1, 2, \ldots, N)$. These particles are propagated to the next time using (48).

$$S_S(n, \tau + 1|\tau) = f(S_S(n, \tau|\tau), C(\tau + 1)), \ n = 1, 2, \ldots, \ N \tag{48}$$

Here, C is the plant noise that is generated randomly based on the PDF of the particle. Once the measurements are obtained, the conditional relative likelihood of each particle, $p(Z(\tau), \ S_S(n, \tau + 1|\tau))$, is computed. For an m-dimensional measurement equation, a relative likelihood $O(\tau)$ can be computed as follows in Equation (49) [12].

$$
\begin{aligned}
O(\tau) &= P[Z(\tau) = Z, S_S(n, \tau) = S_S(n, \tau + 1|\tau)] \\
&= P[\gamma(\tau) = Z - H(S_S(n, \tau + 1|\tau))] \\
&\sim \left(1/(2\pi)^{m/2}\left(\sigma_B^2\right)^{m/2}\right) \times exp\left[\frac{\left(-[Z - H(S_S(n, \tau+1|\tau))]^T[Z - H(S_S(n, \tau+1|\tau))]\right)}{2\sigma_B^2}\right]
\end{aligned}
\tag{49}
$$

'$\sim$' symbol in (47) represents the proportionality relation rather than the equality relation between the probability and expression on the right side. Now the relative likelihoods obtained in (46) are normalized as follows in Equation (50).

$$O(\tau) = O(\tau)/\left(\sum_{i=1}^{N} O(i)\right) \tag{50}$$

Now, the particles using the computed likelihoods are re-sampled. This means a new set of particles are randomly generated based on the relative likelihoods $O(\tau)$. As the number of particles increases to infinity, the accuracy in estimation increases and thereby complexity of computation. At any point in time, these sets of particles can approximate the PDF of the state accurately [21]. The advanced microprocessors, available these days can easily manage the complexity in computation regarding PF.

It is a known fact that PF has the disadvantage of high computational complexity, but this disadvantage can be reduced by the utilization of fast computing processors with high storage memory (which are available now a days). PF also has the disadvantage of sample degeneracy and sample impoverishment. This is reduced by several resampling techniques available in literature [9,22]. However, these techniques are not suitable for all types of applications. Based on application, proper sampling technique must be selected.

One approach that has been proposed for improving particle filtering is to combine it with another filter such as the EKF, UKF, or MGBEKF [12]. Here, each posterior particle will be updated using the posterior updating equations of EKF, UKF, or MGBEKF algorithms and then re-sampling (if required) is performed. This is like working with an N number of Kalman filters (one for each particle), that is initialized with randomly generated state vectors and then re-sampling them (if required) after updating them with each measurement obtained. In this research work, PF combined with the EKF algorithm is explained for fused measurement data in the form of a flowchart in Figure 2. Similarly, PF is combined with MGBEKF. In this paper, the particle filter is combined with the EKF, MGBEKF and the algorithms are named as particle filter coupled with extended Kalman filter (PFEKF) and particle filter coupled with modified gain bearings-only extended Kalman filter (PFMGBEKF). $S_S(n, \tau + 1|\tau)$ is updated to $S_S(n, \tau + 1|\tau + 1)$ according to the EKF and MGBEKF equations [12,13] in PFEKF and PFMGBEKF respectively.
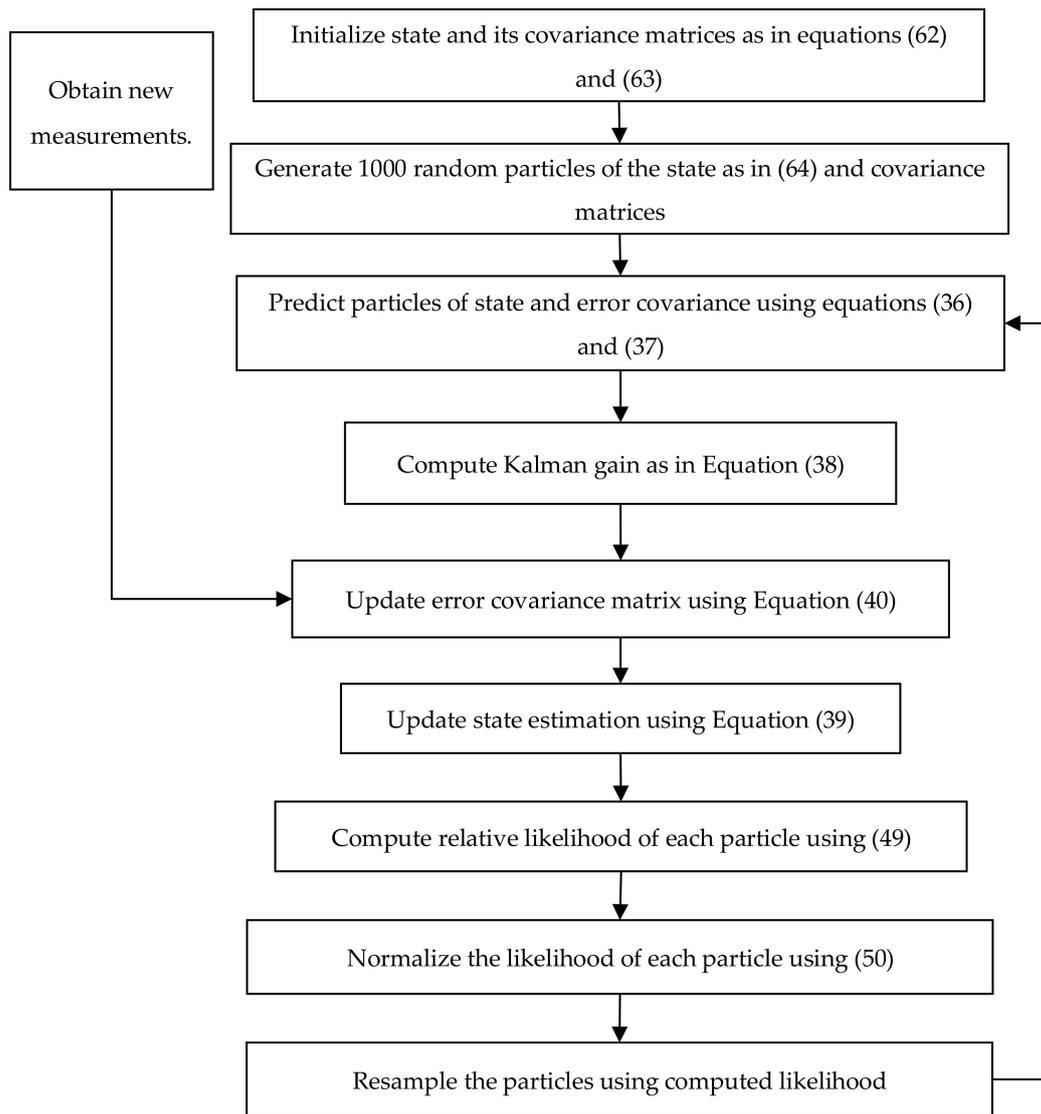
Figure 2. Flow chart representing PFEKF algorithm.

2.6. 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 [21]. It is defined that for a nonlinear discrete-time system, the covariance of the estimated state follows the condition in Equation (51).

$$E\left[\left(\hat{S}_S(\tau) - S_S(\tau)\right)\left(\hat{S}_S(\tau) - S_S(\tau)\right)^T\right] \geq J_\tau^{-1}, \tag{51}$$

where J_τ is an information matrix, which is calculated recursively [11], using Equation (52).

$$J_\tau = D_{\tau-1}^{22} - D_{\tau-1}^{21}\left(J_{\tau-1} + D^{11}\right)^{-1}D_{\tau-1}^{12}, \tag{52}$$

where D^{11}, D^{12}, D^{21} *and* D^{22}, for additive Gaussian noise, are given as follows in Equations (53)–(56).

$$D^{11} = A^T Q^{-1} \tag{53}$$

$$D^{21} = -A^T Q^{-1} \tag{54}$$

$$D^{12} = \left(D^{21}\right)^{T} \tag{55}$$

$$D^{22}_{\tau-1} = Q^{-1} + E\left[H_{\tau}^{T}\left(\sigma_{B}^{2}\right)^{-1}H_{\tau}\right] \tag{56}$$

Here H_{τ} is given by the Jacobian of the measurement function for TST as in Equation (57).

$$H_{\tau} = \begin{bmatrix} 0 & 0 & 0 & 0 & \cos\hat{\beta}_{ham}/R & -\sin\hat{\beta}_{ham}/R \\ 0 & 0 & 0 & 0 & \cos\hat{\beta}_{tam}/R & -\sin\hat{\beta}_{tam}/R \end{bmatrix} \tag{57}$$

and $\hat{\beta}_{ham}$ is the predicted HMA measurement and $\hat{\beta}_{tam}$ is the predicted TA measurement. For SST the equations are as given in [23]. The CRLB of range error is calculated for TST as in Equation (58).

$$\text{CRLB\_R}_{\tau} = \sqrt{J_{\tau}^{-1}(5,5) + J_{\tau}^{-1}(6,6)} \tag{58}$$

Similarly, the CRLB of speed error for TST is defined as in Equation (59).

$$CRLB\_S_{\tau} = \sqrt{J_{\tau}^{-1}(3,3) + J_{\tau}^{-1}(4,4)} \tag{59}$$

CRLB of course error for TST is calculated as in Equation (60).

$$CRLB\_C_{\tau} = \tan^{-1}\left(\sqrt{J_{\tau}^{-1}(3,3)} / \sqrt{J_{\tau}^{-1}(4,4)}\right) \tag{60}$$

3. Implementation and Simulation

Scenarios that are considered for evaluating the algorithms are as shown in Table 1. The bearing measurements from HMA and TA are assumed to be available continuously for every second for TST and SST.

Table 1. Scenarios for the evaluation of algorithms.

Parameter	Scenario No.		
	1	2	3
Initial Range (m)	3000	3000	3000
Initial Bearing (deg)	0	0	0
Target Speed (m/s)	12	12	12
Observer Speed (m/s)	8	8	8
Target course (deg)	155	145	110
Observer course (deg)	90	90	90

In an underwater environment, a submarine in a passive mode of operation can track a target at 4 to 5 kilometers. However, this changes with the environmental conditions underwater. Scenarios with different target courses are considered to track the target moving in different directions. Scenario with an initial bearing of 20° is like the scenario with an initial bearing of 0° with coordinates frame turned by 20°. Hence the bearing value is chosen as 0° for simplicity. Target is also assumed to be a submarine, so, the speed of the target is chosen accordingly [24].

For SST, the observer is presumed to maneuver following the famous S-maneuver in its course. Therefore, for two minutes, the observer follows the current course it is i.e, 90°. Now it slowly turns 180° at the rate of 0.5° per second to attain 270°. With 0.5° per second turn rate, the observer takes 6 minutes to complete the maneuver. For TST, the observer is presumed to be traveling at a constant speed and with a constant course of 90° i.e., without any maneuver in course or speed. The noise in measurements is assumed to follow Gaussian distribution and is additive. The initializations of the target state vector and its covariance matrices are chosen as in Equations (33)–(35). The speed

of the target is difficult to estimate and is assumed to be 5m/s, as the submarine travels at very low speeds. Sonar range of the day, which gives the minimum range of target detection with a 50 percent probability of detection, of 5000 m is considered in target state vector initializations.

For SST, the initial state vector is given in Equation (61) and for TST it is given in Equation (62).

$$S_s(0|0) = [5 \quad 5 \quad 5000\sin\beta_m \quad 5000\cos\beta_m]^T \tag{61}$$

$$S_S(0|0) = [0.1 \quad 0.1 \quad 5 \quad 5 \quad 5000\sin(\beta_{ham}(1)) \quad 5000\sin(\beta_{ham}(1))]^T \tag{62}$$

Initial target state estimate $S(0|0)$ is assumed to be distributed uniformly and elements of the state covariance matrix are initialized accordingly as Equation (63).

$$P(0|0) = Diagonal\left(4S_S^2(0|0)/12\right)_i \tag{63}$$

Here $i = 1, 2, \ldots, 6$ for TST and $i = 1, 2, \ldots, 4$ for SST

The particles of the target state vector for PFEKF and PFMGBEKF filters are generated by adding random noise to the initial state vector as in (58) for TST. Similarly, for SST the initial state vector is added with a column matrix of random numbers with four rows as in Equation (64) [22].

$$(S_S(0|0))_j = S_S(0|0) + \begin{bmatrix} randn & randn & randn & randn & randn & randn \end{bmatrix}^T \tag{64}$$

Here, j represents the particle number and $j = 1, 2, 3, \ldots, 1000$. The covariance matrix for each particle of the target state vector for PFEKF and PFMGBEKF filters are generated based on each particle generated.

It is not genuine to evaluate the performance of the algorithm on a run basis, as randomness exists in the experiment. Therefore, Monte Carlo simulation of 100 runs is performed for each filtering in MATLAB [6] for both SST and TST algorithms. The maximum (3σ) error acceptable in estimated range, course and speeds are 8% of true range, $3°$ and 1m/s, the root-mean-squared (RMS) (1σ) error allowed for the acceptance of the solution are 2.66% of true range, $1°$ and 0.33m/s respectively. The errors in the estimation of target parameters are root-mean-squared and the solution is assessed on the error acceptance levels mentioned above.

RMSE is nothing but root mean squared error and it is, in general, called standard deviation. It is implemented as follows. The difference between the true target parameter and the estimated target parameter is taken as error and it is squared. The mean of the squared error for all the 100 runs is then calculated and square rooted. This provides the RMSE of the estimated target parameters. Let i be the number of Monte Carlo runs and j be the total time in seconds for which the simulation is carried out. Then the error is mathematically given as Equation (65) and the RMSE as Equation (66).

$$E_{ij} = (true\ target\ parameter)_j - (estimated\ target\ parameter)_{ij} \tag{65}$$

$$(RMSE)_j = \sqrt{mean\left(\left(E_{ij}\right)^2\right)} \tag{66}$$

The error in the estimated target parameters must be within the vicinity range of the weapon that has to be fired onto the target. The error in estimated target parameters is calculated based on Monte Carlo runs. Therefore, the RMSE values of the estimated target parameters are considered. With Monte Carlo simulations with 100 different noise sequences (maintaining same PDF), the outputs are root mean squared to obtain a single value. The acceptance criteria are chosen based on the weapon homing zone. The solution is said to be obtained only when the error in the estimated target parameters is less than the given values in acceptance criteria. Once the solution is obtained, the estimated target parameters can be used to release weapon onto the target.

The convergence time of the solutions for the three scenarios based on the above-mentioned acceptance criteria for 100 runs is tabulated in Table 2 for different filtering algorithms. In a Monte Carlo simulation, it is observed that UKF converges faster than other filters for BOT. Moreover, with single bearing measurement, the observer needs to follow S-maneuver, which can be avoided while using bearing measurements from two sensors.

Table 2. Convergence times of target parameters in seconds using the SST algorithm.

Filter	Scenario No.	Target Parameter				Simulation Time
		R	C	S	TC	
EKF	1	273	301	330	330	462.82
	2	NC	NC	NC	NC	239.49
	3	339	416	339	416	408.42
MGBEKF	1	NC	NC	NC	NC	287.53
	2	NC	NC	NC	NC	371.97
	3	249	321	240	321	380.34
UKF	1	216	216	216	216	329.3
	2	244	239	329	329	345.6
	3	320	338	318	338	441.23
PFEKF	1	247	247	247	247	263.85
	2	249	246	245	249	122.39
	3	343	434	340	434	236.39
PFMGBEKF	1	273	320	269	320	398.29
	2	236	337	301	337	389.29
	3	270	457	288	457	228.24

R: Range; C: Course; S: Speed; TC: Total Convergence; NC: No Convergence.

The simulation is carried out for 30 minutes. Results obtained with UKF for scenario 1 in Table 1 with SST and TST, are presented in Figures 3 and 4 respectively. For SST, the observer needs to perform S-maneuver to obtain the observability of the process and hence to obtain a solution. With TST, the observability and solution are obtained even without observer maneuver as it uses bearing measurements from two sensors.
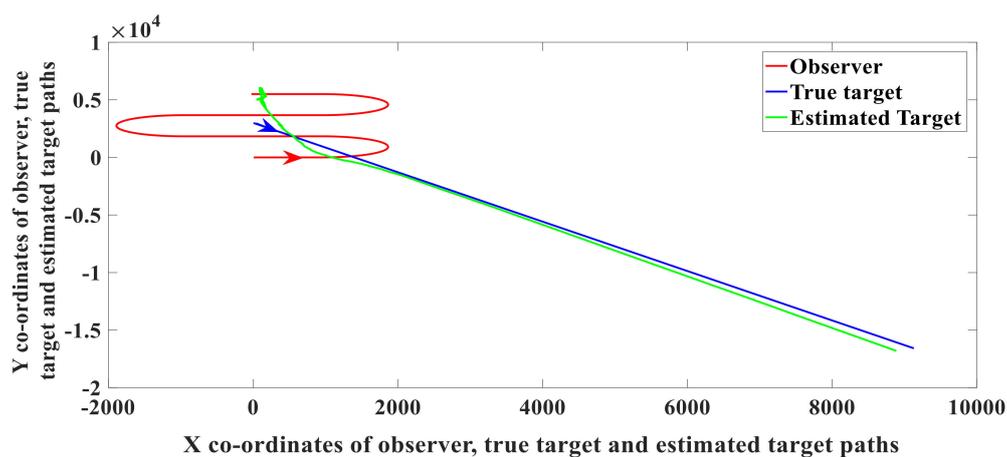
Figure 3. Target and observer paths with SST for scenario 1.
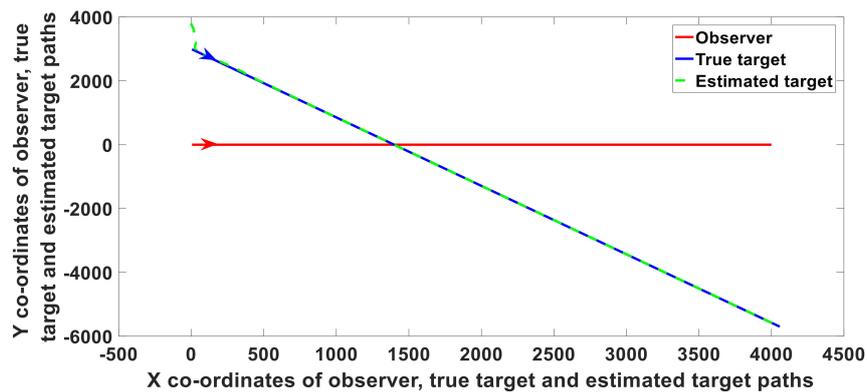
Figure 4. Target and observer paths with TST for scenario 1.

Table 3 gives the total convergence times of the solution with various filtering algorithms for TST. The solution is said to be converged if the acceptance criteria are achieved in all the three parameters at a time and the solution does not diverge for the rest of the simulation time. It can be observed from the table data that the solution convergence time is only for a short period with some of the scenarios and not for the entire simulated time for EKF. For scenario 2, the solution is converged from 227 to 447th second and then the solution diverged for the rest of the simulation time. The solution is not obtained for some of the scenarios with EKF. With MGBEKF, the same trend of solution-divergence persists. With UKF, the solution is obtained earlier than other filters for most of the scenarios. As shown in Table 3, UKF for scenario 1 has convergence time of 235 seconds whereas PFEKF and PFMGBEKF have convergence times of 337 and 237 seconds respectively. It can also be observed from Tables 2 and 3 data that the convergence times for TST are obtained faster than SST.

Table 3. Convergence times of target parameters in seconds using the TST algorithm.

Filter	S. No	Target Parameter				Simulation Time
		R	**C**	**S**	**TC**	
EKF	1	155	200–253; 347–442	213–257; 301–532	200–253; 347–442	301.81
	2	158–447	227–447	182–447	227–447	255.68
	3	109–148; 187–828	NC	226–676	NC	441.86
MGBEKF	1	18–676; 679–730	206–296; 377–401	159–357	206–296; 386–410	1160.08
	2	17–296	NC	124–296	NC	315.04
	3	218	333–365; 465–485; 550–611; 655–675	206–246; 260–280	NC	385.01
UKF	1	193	235	230	235	427.36
	2	175	248	221	248	430.81
	3	202	333	140	333	325.57
PFEKF	1	116	337	242	337	141.40
	2	119	437	259	437	159.59
	3	114	713	200	713	198.75
PFMGBEKF	1	116	237	150	237	436.61
	2	119	256	152	256	1038.03
	3	130	712	85	712	163.65

S. No: Scenario Number; R: Range; S: Speed; C: Course; TC: Total Convergence; NC: No Convergence.

As the measurements are taken from two different sensor arrays and both the measurements are nonlinearly related to the target state vector elements, the process nonlinearity will be very high. Therefore, sub-optimal nonlinear filters, EKF and MGBEKF, failed to give proper solutions. Particle filter deals with complete non-linearity and is suitable for highly nonlinear systems. However, to avoid the sample degeneracy problem it is combined with other filters and hence gives better results than EKF and MGBEKF.

RMS errors in range, course, and speed estimates of the target are represented in Figures 5–7. It can be observed from the figures that the RMS errors in estimated target parameters are less in case of TST than that with SST. RMS errors in estimated parameters are obtained within the acceptance criteria for TST much earlier than that of SST. However, as the range between observer and target increases, the RMS errors in estimated target parameters increases slightly for TST than that of SST.
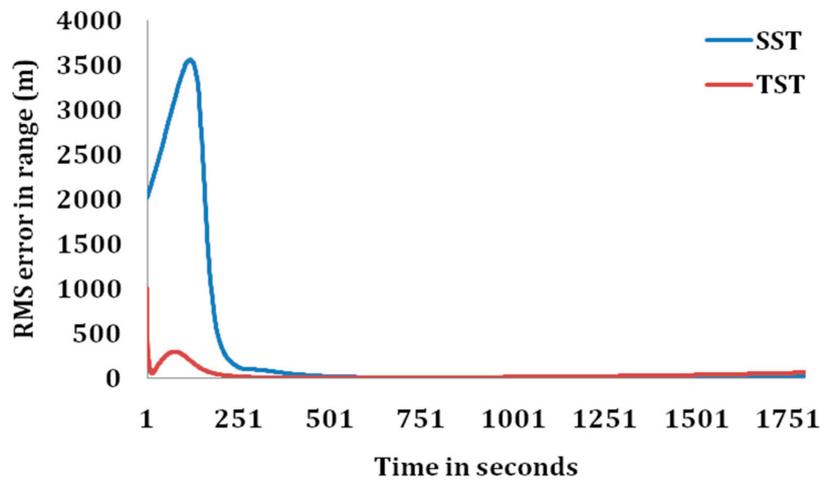
Figure 5. RMS error in range estimates of the target with SST and TST.
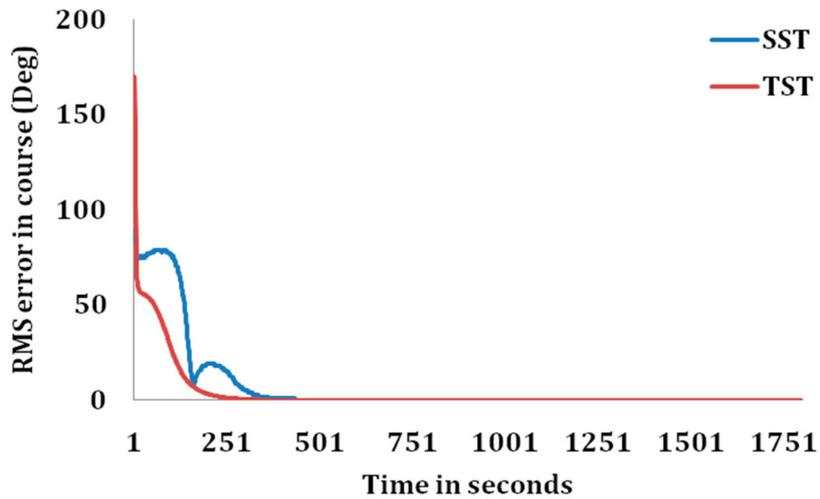
Figure 6. RMS error in course estimates of the target with SST and TST.
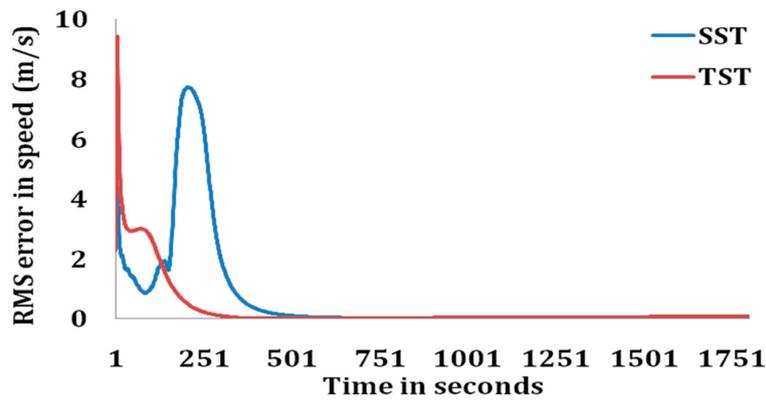
Figure 7. RMS error in speed estimates of the target with SST and TST.

Figure 8 shows the acceptable error in range along with CRLB values for scenario 3, according to the acceptance criteria mentioned above, which is 2.667% of true range. RMS errors in the range of the target for different filters are also shown in Figure 8. It can be observed from the figure that lower RMS error is obtained earlier with PFEKF and PFMGBEKF filters than that of EKF, MGBEKF, and UKF filters. It can be seen from Figure 8b that the lowest RMS error is obtained with PFEKF. Though the error is below the acceptable error values, there is an effect of EKF that makes the error to vary in an unstable manner, which can be seen in the form of spikes in the plot. Similarly, with PFMGBEKF, the instability in error is reduced when compared to PFEKF.

(**a**)

(**b**)
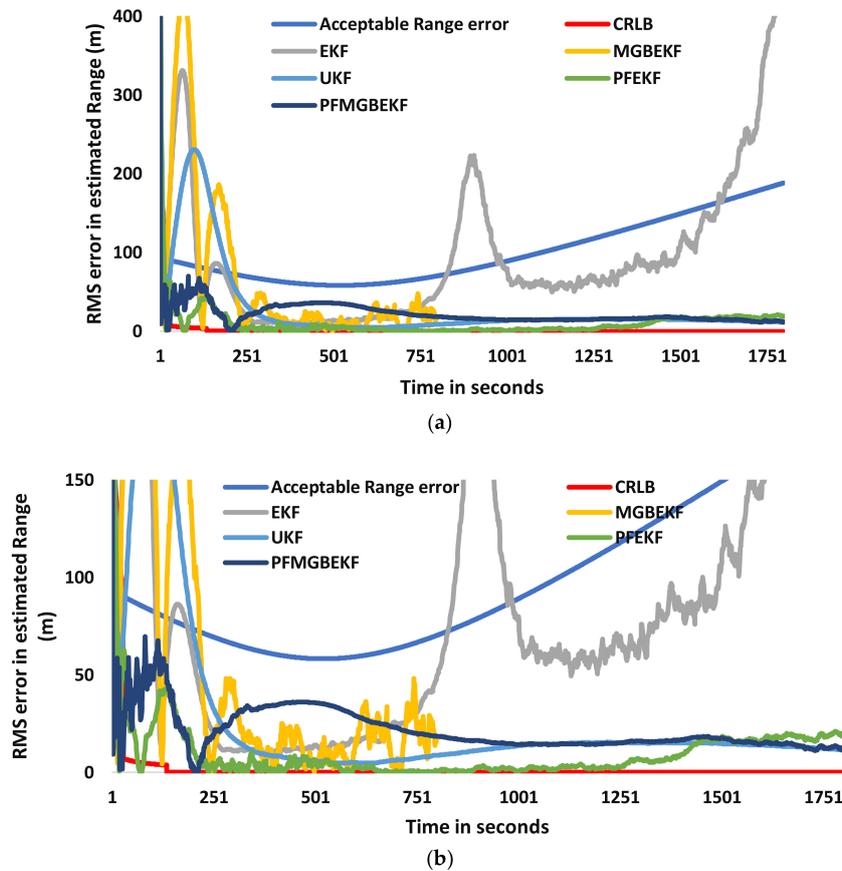
Figure 8. RMS error in range estimate of the target for scenario 3. (**a**) RMS error in range estimate of the target. (**b**) Expanded view of Figure 8a.

Figure 9 shows the acceptable error in the course for the same scenario 3, according to acceptance criteria mentioned above, which is less than 1°. RMS errors in the course of the target for different filters are also shown in Figure 9. From Figure 9a, it can be observed that the error in the course is not obtained within acceptance criteria with EKF and MGBEKF. However, the fastest convergence in the target course was observed to be obtained with UKF, which can be seen from Figure 9b. PFEKF and PFMGBEKF also have convergence in the target course at 562 and 550 s respectively.
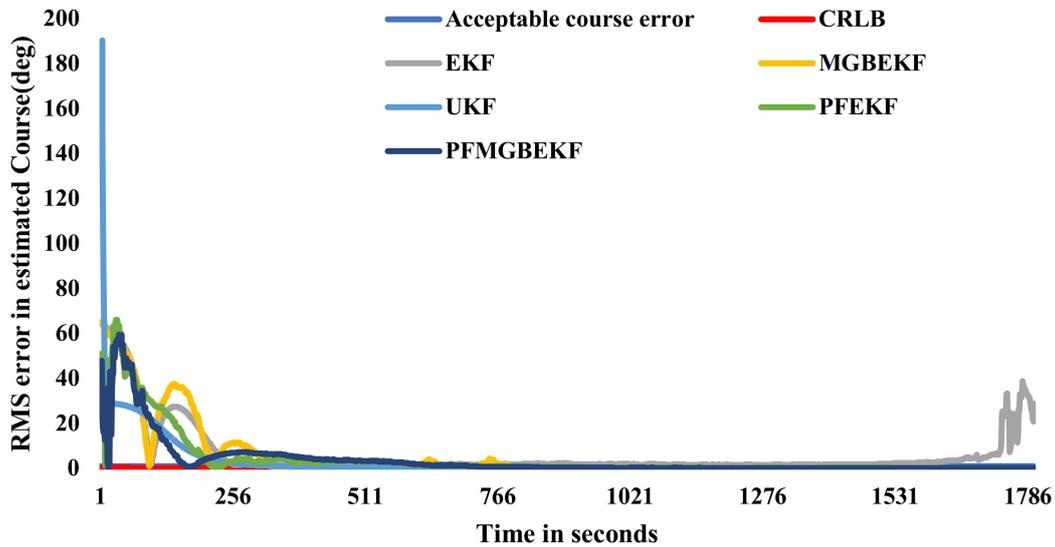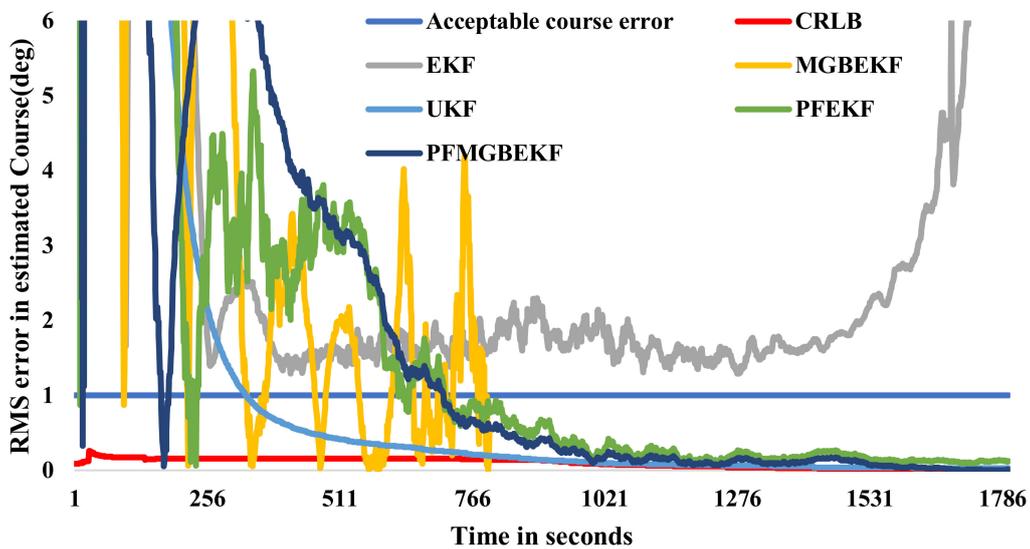
Figure 9. RMS error in course estimate of the target for scenario 3. (**a**) RMS error in course estimate of the target. (**b**) Expanded view of Figure 9a.

Figure 10 shows the acceptable error in speed for the same scenario 3, according to the acceptance criteria mentioned above, which is less than 0.33 m/s. RMS errors in the speed of the target for different filters are also shown in Figure 10. It can be observed from the figure that RMS error in speed with EKF and MGBEKF are varying in an unstable manner and are out of bounds after a certain time. With UKF, the convergence of speed error within acceptable criteria takes a shorter time than other filters and has

a stable pattern. With PFMGBEKF, the lowest values in error are obtained. The overall convergence times were observed to be shorter for UKF than other filters and hence UKF performs well for TST than other filters.
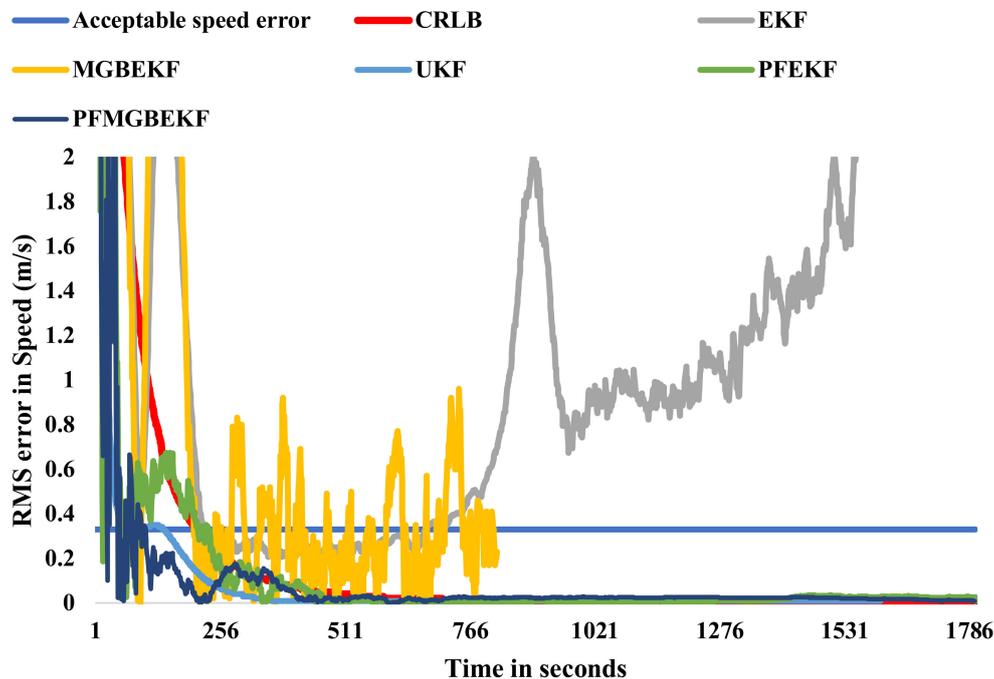
Figure 10. RMS error in speed estimate of the target.

4. Conclusions

In an underwater scenario, the observer and target are considered as submarines. Accordingly, scenarios are considered covering different target courses. SST and TST algorithms are considered for the performance evaluation concerning the convergence of the solution using different filters. Simulation is carried out and it is confirmed that TST generates the solution faster when compared to SST and even without observer maneuver. Hence, for passive target tracking, tracking with two bearing measurements is recommended. For TST, it was observed from simulated results that UKF provides a faster solution than other filtering algorithms.

Author Contributions: Conceptualization, K.J. and K.R.S.; Methodology, K.J. and K.R.S.; Software, K.J.; Validation, K.J. and K.R.S.; Formal analysis, K.J. and K.R.S.; Writing—original draft preparation, K.J. and K.R.S.; Writing—review and editing, K.R.S.; Supervision, K.R.S. All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nardone, S.C.; Lindgren, A.G.; Gong, K.F. Fundamental properties and performance of conventional bearings-only target motion analysis. *IEEE Trans. Autom. Control* **1984**, *29*, 775–787. [CrossRef]
2. Lindgren, A.G.; Gong, K.F. Position and velocity estimation via bearing observations. *IEEE Trans. Aerosp. Electron. Syst.* **1978**, *14*, 564–577. [CrossRef]
3. Aidala, V.J. Kalman filter behavior in bearings-only tracking applications. *IEEE Trans. Aerosp. Electron. Syst.* **1979**, *15*, 29–39. [CrossRef]

4. Lindgren, A.G.; Gong, K.F. Properties of a bearings-only motion analysis estimator: An interesting case study in system observability. In Proceedings of the 12th Asilomar Conference on Circuits, System, and Computing, San Diego, CA, USA, 6–8 November 1978; pp. 265–272.

5. Nardone, S.C.; Aidala, V.J. Observability criteria for bearings-only target motion analysis. *IEEE Trans. Aerosp. Electron Syst.* **1981**, *17*, 162–166. [CrossRef]

6. Koteswara Rao, S. Pseudo-linear estimator for bearings-only passive target tracking. *IEEE Proc. Radar Sonar Navig.* **2001**, *148*, 16–22. [CrossRef]

7. Wan, E.A.; Van Der Merwe, R. The unscented Kalman filter for nonlinear estimation. In Proceedings of the IEEE Symposium 2000 on Adaptive Systems for Signal Processing, Communication and Control, Lake Louise, AB, Canada, 1–4 October 2000; pp. 153–158. [CrossRef]

8. Julier, S.J.; Uhlmann, J.K. A new extension of the Kalman filters to nonlinear systems. In *Signal Processing, Sensor Fusion, and Target Recognition VI*; International Society for Optics and Photonics: Bellingham, WA, USA, 1997; pp. 182–193.

9. Koteswara Rao, S.; Raja Rajeswari, K.; Linga Murthy, K.S. Application of unscented Kalman filter for tracking underwater targets using bearings only measurements. In Proceedings of the IEEE International Conference on Soft Computing and Machine Learning for Signal Processing, Control, Power and Telecommunications, Bhubaneswar, India, March 2006; pp. 181–184.

10. Koteswara Rao, S.; Raja Rajeswari, K.; Linga Murthy, K.S. Unscented Kalman filter with application to bearings-only target tracking. *IETE J. Res.* **2009**, *55*, 63–67. [CrossRef]

11. Ristick, B.; Arulampalam, M.S.; Gordon, N. *Beyond Kalman Filters–Particle Filters for Tracking Applications*; Artech House, DSTO: Boston, London, 2004.

12. Simon, D. *Optimal State Estimation: Kalman, H∞ and Nonlinear Approximations*; Wiley: Hoboken, NJ, USA, 2006.

13. Bar-Shalom, Y.; Fortmann, T.E. *Tracking and Data Association*; Academic Press Inc.: New York, NY, USA, 1998.

14. Koteswara Rao, S.; Raja Rajeswari, K.; Linga Murthy, K.S. Data fusion for underwater target tracking. *IET Radar Sonar Navig.* **2009**, *4*, 576–585. [CrossRef]

15. Atherton, D.P.; Bather, J.A.; Briggs, A.J. Data fusion for several Kalman filters tracking a single target. *IEEE Proc. Radar Sonar Navig.* **2005**, *152*, 372–376. [CrossRef]

16. Liu, G.; Wang, Y. Multi-Sensor Moving Target Tracking Using Particle Filter. In Proceedings of the IEEE International Conference on Robotics and Biomimetics, Sanya, China, 15–18 December 2007; pp. 668–673. [CrossRef]

17. Yang, R.; Bar-Shalom, Y.; NG, G.W. Bearings-Only Tracking with Fusion from Heterogenous Passive Sensors: ESM/EO and Acoustic. *J. Adv. Inf. Fusion* **2017**, *12*.

18. Lee, E.H.; Song, T.L. Multi-sensor track-to-track fusion with target existence in cluttered environments. *IET Radar. Sonar Navig.* **2017**, *11*, 1108–1115. [CrossRef]

19. Dong, Y.; Zhang, G.; He, X.; Tang, J. Information Fusion in Networked Underwater Target Detection. In Proceedings of the IEEE OCEANS, Genoa, Italy, 18–21 May 2015; pp. 1–4. [CrossRef]

20. Li, Y.; Zhao, Z. Passive Tracking of Underwater Targets Using Dual Observation Stations. In Proceedings of the International Bhurban Conference on Applied Sciences & Technology (IBCAST), Islamabad, Pakistan, 8–12 January 2019; pp. 867–872. [CrossRef]

21. Kausar, J.; Koteswara Rao, S. Implementation of underwater target tracking techniques for Gaussian and non-Gaussian environments. *Comput. Electr. Eng.* **2020**, *87*, 106783. [CrossRef]

22. Kausar, J.; Koteswara Rao, S. Extended Kalman Filter for Bearings-Only Tracking. *Int. J. Eng. Adv. Technol. (IJEAT)* **2019**, *8*, 637–640. [CrossRef]

23. Li, T.; Bolić, M.; Djurić, P.M. Resampling Methods for Particle Filtering: Classification, implementation, and strategies. *IEEE Signal Process. Mag.* **2015**, *32*, 70–86. [CrossRef]

24. Koteswara Rao, S. Bearings-Only Tracking. Observer Maneuver Recommendation. *IETE J. Res.* **2018**. [CrossRef]