Optimization Algorithm of Bearings Only Tracking based on Cubature Kalman Filter

Fu Xiao1,2,3, a, Liu Lei1,2,3, b

1Institute of Oceanographic Instrumentation, Qilu University of Technology (Shandong Academy of Sciences).
2Shandong Provincial Key Laboratory of Ocean Environmental Monitoring Technology.
3National Engineering and Technological Research Center of Marine Monitoring Equipment Qingdao China

Corresponding author e-mail: maiden_fu@163.com, bliulei_private@163.com.

Abstract. In view of the shortcomings of low precision and poor performance in the process of bearings only tracking (BOT) in wireless sensor networks (WSNs), an optimization algorithm of BOT based on the cubature Kalman filter (CKF) in WSNs is proposed. The sensors with spatial distribution are divided into several clusters, the number of sensors in each cluster is the same, and each sensor detects data to form a local estimation. Then uploads it to the cluster head (CH) node data fusion, so as to get the accurate information of the target. In order to improve the positioning accuracy, the CKF is used to process the data of each sensor, and then it is uploaded to the CH node to ensure the tracking ability of the filter to the changes of structural parameters. The CH node uses weighted data fusion algorithm to get the final fusion result. The simulation results verify the effectiveness of the method.

1. Introduction

Bearings only tracking (BOT) is widely used in navigation, passive tracking and so on, among which only bearings are used for target location or tracking. With the rapid development of sensor, microprocessor, wireless communication and network technology, BOT tracking has made significant progress [1-5]. Target tracking is one of the most representative and basic applications in wireless sensor networks (WSNs) [6]. WSNs is a kind of distributed sensor network, its terminal is the sensor that can sense and check the external world. WSNs are composed of many low-cost and low-power sensors, which are used to coordinate target tracking [8]. A typical sensor network consists of a group of sensor nodes distributed in a specific monitoring area [9]. Considering the problem of accuracy, in practical application, only one sensor cannot meet the actual work needs, so it is necessary to increase the number of sensors, and fuse the data of multiple sensors in space [10, 11]. However, if the space is large and the number of sensors is large, communication between each sensor and the fusion center will cause waste of resources, and it is impossible to allocate communication channels to each sensor [12]. In order to solve this problem, a
clustering fusion algorithm is proposed [13, 14]. Clustering fusion algorithm can reduce network congestion and improve the reliability and robustness of WSNs. The main idea of clustering fusion is to use local estimator for each intelligent sensor to disperse the function of fusion center [15, 16].

The filtering algorithm has experienced the extension from linear filtering algorithm to nonlinear filtering algorithm, which is also the result of practical engineering. Among the filtering algorithms of linear system, Kalman filter (KF) based on state space method is the most widely used. KF is a time-domain filter [15]. In the nonlinear filtering algorithm, the first application is extended Kalman filtering (EKF), which is based on the idea of the first-order Taylor series expansion of the nonlinear function, approximates the nonlinear system to a linear system, and then uses KF to process [13]. However, due to the shortcomings of EKF algorithm, the application of EKF in engineering practice is limited. Cubature Kalman filtering (CKF) is based on the third-order spherical radial cubature criterion, and uses a group of cubature points to approximate the mean and covariance of the state of the nonlinear system with additional Gaussian noise [5]. It is the closest approximation algorithm in theory to Bayesian filtering, and it is a powerful tool to solve the state estimation of the nonlinear system. In [5], this paper proposes the research of the CKF algorithm and its application in navigation. It proposes that the CKF selects a group of point sets through certain strategies, transfers the selected point sets through nonlinear functions, and then approximates the Gauss integral in the nonlinear Gauss filter through corresponding calculation. It avoids the linearization of nonlinear functions by EKF algorithm, and can deal with any problem in the process of filtering, it does not depend on the specific form of nonlinear function and does not need to calculate Jacobian matrix. In [4], the application of the CKF algorithm in SINS/BDS integrated navigation is proposed. In [3], an adaptive feedback CKF algorithm is proposed, which is used in BOT tracking. The structure of clustering fusion estimation of sensor network is shown in Fig. 1.

Based on the above analysis, an optimization algorithm of BOT based on CKF is proposed. The sensors are divided into several same clusters. Each cluster of sensors collects data and uploads it to the cluster head (CH) node to track the target using the azimuth angle. In the nonlinear filtering, the weight of each accumulation point of CKF algorithm is positive, so there is no problem that the covariance is not positive due to the weighted sum.

Fig 1. The structure of clustering fusion estimation of sensor network.

2. Problem formulation
The state space model of this paper considers the constant speed model of two-dimensional BOT tracking. In order to save energy consumption and improve the utilization rate of sensor resources, clustering method is adopted in this paper. Fig. 2 depicts the measurement of angles in a Cartesian coordinate system. The equation of state is as follows.
Fig 2. Diagram of BOT.

\[ X(t_{k+1}) = AX(t_k) + \omega(t_k) \]  
(1)

\[ X(t_k) = [x(t_k), \dot{x}(t_k), y(t_k), \dot{y}(t_k)] \]  
(2)

Where \( x(t_k) \) and \( y(t_k) \) are the locations of the target on the \( x \) axis and \( y \) axis. \( \dot{x}(t_k) \) and \( \dot{y}(t_k) \) are the velocity of the target on the \( x \) axis and \( y \) axis. \( A \) is time-varying matrices of appropriate dimensions. \( \omega(t_k) = [\omega_x(t_k), \omega_y(t_k)]^T \) is white Gaussian random variables with zero mean values and satisfy.

\[
E[\omega(t_k)] = 0 \\
E[\omega(t_k)\omega^T(t_k)] = Q(t_k)
\]  
(3)

The measurement equation is as follows:

\[
Z(t_{k,m,i}) = h(X(t_{k,m,i})) + v(t_{k,m,i}) \\
i = M_i, m \in Z_0
\]  
(4)

Where \( Z(t_{k,m,i}) \) represents the measured azimuth value, \( h(X(t_{k,m,i})) \) represents the non-linear measurement function expression, i.e. the true azimuth. \( Z_0 \in \{1, 2, ..., m\} \) represents the set of clusters. It needs to be explained that the actual azimuth is easily polluted by noise, and the measured azimuth is not the true target azimuth but the sum of \( h(X(t_{k,m,i})) \) and \( v(t_{k,m,i}) \). \( h(X(t_{k,m,i})) \) and \( v(t_{k,m,i}) \) are as follows.

\[
h(X(t_{k,m,i})) = \arctan\left(\frac{y(t_{k,m,i}) - y_0}{x(t_{k,m,i}) - x_0}\right)
\]  
(5)
In the formula, $x(t_{k,m,i})$ and $y(t_{k,m,i})$ represent the components of the sensor in the x-axis and y-axis, $x_0$ and $y_0$ are the components of the target in the x-axis and y-axis, and the sensor node is used to detect the output of the system (1). $v(t_{k,r,i})$ is zero mean Gaussian white noise. $o(t_k)$ and $v(t_{k,r,i})$ are mutually independent and the covariance is as follows.

$$E[v(t_{k,m,i})] = 0$$

$$E[v(t_{k,m,i})v^T(t_{k,m,i})] = R(t_{k,m,i})\delta_{ij}\delta_{ks}$$

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

$$\delta_{ks} = \begin{cases} 1, & k = s \\ 0, & k \neq s \end{cases}$$

3. **Fusion estimation optimization algorithm**

Cluster fusion estimation is divided into two parts. Sensor nodes can communicate with each other. The collected measurement values are formed into local estimation. CKF is used for local estimation. Matrix weighted data fusion algorithm is used to fuse CH nodes, and the final fusion result is obtained and the clustering fusion estimation is carried out. The sequence analysis process diagram is shown in Fig.3.

![Fusion estimation time series analysis process diagram](image)

**Fig 3.** Fusion estimation time series analysis process diagram.

### 3.1. CKF Design

Because the measurement equation is nonlinear, each sensor in WSNs needs a nonlinear estimation filter to estimate the position. In $M\epsilon$, the local estimation interval is $[t_{k-1,m,i}, t_{k,m,i}]$. The CKF is used to estimate the optimal state of each sensor node.

1) The initial state and initial error covariance matrix are as follows.

$$E[X(0)] = X_0$$

$$E[(X(0) - X_0)(X(0) - X_0)^T] = P_0$$

2) Time update: For the recursive formula of nonlinear Gaussian filtering, if it is to be converted into a specific realizable filtering formula, it needs various approximation strategies, and the implementation steps of CKF algorithm based on the third-order spherical phase diameter volume rule are as follows.
Assuming that the statistical property, the state $X(t_{k-1,m,i}/t_{k-1,m,i})$ at k-1 time is known, first make the Cholesky decomposition for $P(t_{k-1,m,i}/t_{k-1,m,i})$, then.

$$
P(t_{k-1,m,i}) = S(t_{k-1,m,i}/t_{k-1,m,i})S^T(t_{k-1,m,i}/t_{k-1,m,i})
$$

(11)

Select the cubature point as.

$$
\hat{X}_\kappa(t_{k-1,m,i}/t_{k-1,m,i}) = S(t_{k-1,m,i}/t_{k-1,m,i})\xi_\kappa + \hat{X}_\kappa(t_{k-1,m,i}/t_{k-1,m,i})
$$

(12)

$$\kappa = 1, 2, ..., \sigma$$

Where $\kappa = 1, 2, ..., \sigma$ is the number of cubature points, Using the third-order spherical radial cubature criterion, the following basic cubature points and corresponding weights are obtained.

$$
\xi_\kappa = \frac{\sqrt{\sigma}}{2} \frac{1}{\kappa}
$$

$$
\omega_\kappa = \frac{1}{\sigma}
$$

(13)

According to the third-order spherical radial cubature criterion, the total number of cubature sampling points is 2 times of the state dimension, i.e. $\sigma = 2N$; $N$ is the system state dimension; [1] represents the $\kappa$th point in the point set [], where the symbol represents the complete full symmetric point set. Calculate the cubature point transferred by the system equation.

$$
\hat{X}_\kappa(t_{k,m,i}/t_{k-1,m,i}) = A\hat{X}_\kappa(t_{k-1,m,i}/t_{k-1,m,i}) + Q_k
$$

(14)

Estimate the state prediction value and prediction error covariance at time k.

$$
\hat{X}(t_{k,m,i}/t_{k-1,m,i}) = \sum_\kappa \omega_\kappa \hat{X}_\kappa(t_{k,m,i}/t_{k-1,m,i})
$$

(15)

$$
P(t_{k,m,i}/t_{k-1,m,i}) = \sum_\kappa \omega_\kappa \hat{X}_\kappa(t_{k,m,i}/t_{k-1,m,i})\hat{X}_\kappa^T(t_{k,m,i}/t_{k-1,m,i}) - \hat{X}(t_{k,m,i}/t_{k-1,m,i})\hat{X}^T(t_{k,m,i}/t_{k-1,m,i}) + Q(t_k)
$$

(16)

3) Measurement update: Cholesky decomposition of $P(t_{k,m,i}/t_{k-1,m,i})$.

$$
P(t_{k,m,i}/t_{k-1,m,i}) = S(t_{k,m,i}/t_{k-1,m,i})S^T(t_{k,m,i}/t_{k-1,m,i})
$$

(17)

Select the cubature point as.
\[
\hat{X}_k(t_{k,m,i} | t_{k-1,m,i}) = S(t_{k,m,i} | t_{k-1,m,i}) \xi_k + \hat{X}(t_{k,m,i} | t_{k-1,m,i})
\]

Transfer cubature point by measuring equation.
\[
Z_k(t_{k,m,i}) = \arctan\left(\frac{y(t_{k,m,i}) - y_0}{x(t_{k,m,i}) - x_0}\right) + R_{k,m,i}
\]

Estimate the observation prediction value and prediction error covariance matrix at k time.
\[
\hat{Z}(t_{k,m,i}) = \sum_{\kappa} a_k Z_k(t_{k,m,i})
\]
\[
P_Z(t_{k,m,i} | t_{k,m,i}) = \sum_{\kappa} a_k Z_k(t_{k,m,i}) Z_k^T(t_{k,m,i}) - \hat{Z}(t_{k,m,i}) \hat{Z}^T(t_{k,m,i}) + R_{k,m,i}
\]

To estimate the Cross-covariance matrix of k time one-step prediction.
\[
P_{XZ}(t_{k,m,i} | t_{k-1,m,i}) = \sum_{\kappa} a_k \hat{X}_k(t_{k,m,i} | t_{k-1,m,i}) Z_k^T(t_{k,m,i}) - \hat{X}(t_{k,m,i} | t_{k-1,m,i}) \hat{Z}^T(t_{k,m,i})
\]

Calculate filter gain.
\[
K = P_{XZ}(t_{k,m,i} | t_{k-1,m,i}) P_Z(t_{k,m,i} | t_{k,m,i})^{-1}
\]

Calculate the k time state estimate and estimate error covariance matrix.
\[
\begin{align*}
\hat{X}(t_{k,m,i} | t_{k,m,i}) &= \hat{X}(t_{k,m,i} | t_{k-1,m,i}) + K(Z(t_{k,m,i}) - \hat{Z}(t_{k,m,i})) \\
\hat{P}(t_{k,m,i} | t_{k,m,i}) &= \hat{P}(t_{k,m,i} | t_{k-1,m,i}) - KP_Z(t_{k,m,i} | t_{k,m,i}) K^T
\end{align*}
\]

3.2. Fusion algorithm Design

Weighted data fusion algorithm is used for fusion estimation. Assuming the weight factor of the sensor is \(\gamma_i\), the following conditions are met.
\[
\sum_{i=1}^{\ell} \gamma_i = 1
\]
The fusion estimate and the estimation error covariance matrix are as follows.

\[
\hat{X}_{m,\text{fusion}} = \sum_{i=1}^{L} \gamma_i \hat{X}(t_{k,m,i} | t_{k,m,i})
\]

\[
P(t_{k,m} | t_{k,m}) = \sum_{i=1}^{L} \gamma_i P(t_{k,m,i} | t_{k,m,i})
\]

\[
\gamma_i = \frac{1}{P(t_{k,m,i} | t_{k,m,i})} \sum_{i=1}^{L} \frac{1}{P(t_{k,m,i} | t_{k,m,i})}
\]

The final fusion results are as follows.

\[
\hat{X}_{m,\text{fusion}} = \sum_{i=1}^{L} \gamma_i \hat{X}(t_{k+1,m,i} | t_{k+1,m,i})
\]

\[
= \sum_{i=1}^{L} \hat{X}(t_{k+1,m,i} | t_{k+1,m,i}) \frac{1}{\sum_{i=1}^{L} P(t_{k+1,m,i} | t_{k+1,m,i})}
\]

4. Simulation and analysis

In order to verify the effectiveness of optimization algorithm for sensor networks with clusters, a non-linear model of two-dimensional space BOT is considered. The description is as follows.

\[
X(t_{k+1}) = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} X(t_k) + w(t_k)
\]

\[
Z(t_{k,m,i}) = \arctan(\frac{y(t_{k,m,i}) - y_0}{x(t_{k,m,i}) - x_0}) + v(t_{k,m,i})
\]

\[
i = M_m, m \in Z_0
\]

In this paper, we fuse the BOT data collected by sensors to prove the effectiveness of the algorithm. The sensors are divided into three clusters, each containing three sensors. Set the noise covariance of the cluster as \(R_{m,i} = 0.1\). The process noise of target tracking is set as \(Q_k = 10^{-3} \times \text{diag}(5,1)\). The simulation number is set to 20. The simulation results are shown in Fig.4-Fig.6. Root mean square error (RMSE) is defined as.

\[
RMSE(t_k) = \left[ \frac{1}{N} \sum_{i=1}^{N} (x_i(t_k) - \hat{x}_i(t_k))^2 \right]^{\frac{1}{2}}
\]
Fig. 4 and Fig. 5 above show the comparison of RMSE of BOT trajectory, local estimation and fusion estimation. Local estimation is the data of single sensor estimation, and fusion estimation is the result of fusion of sensor data. It can be seen from the two figures that the fusion estimation trajectory is closer to the real trajectory than the real one. Fig. 5 compares the results of local estimation and fusion estimation from the angle of error. It can be seen that the root mean square error of local estimation is smaller than that of fusion estimation. Therefore, the accuracy of fusion estimation is higher than that of local estimation.

Fig 4. BOT trajectory comparison.

Fig 5. Local estimation and fusion estimation RMSE.
Fig. 6 shows the comparison between KF and CKF. The proposed CKF has more advantages than the traditional KF. It can be seen from the figure that the proposed CKF error is less than the RMSE of KF, so the accuracy is high.

5. Conclusion
In this paper, an optimization algorithm of BOT based on CKF is proposed to solve the problems of low estimation accuracy and poor performance. The CH node uses weighted data fusion algorithm to get the final fusion result. Simulation results show that BOT filtering is effective. The accuracy of this method is higher than that of KF, and the effect of fusion estimation is more obvious than that of local estimation.

Acknowledgment
This project is supported by Shandong provincial key research and development program (Grant No. 2019GHY112056).

References
[1] H. Wu, S. Chen, X. Y. H. Zhang and B. F. Yang, Robust Bearings-Only Tracking Algorithm using Structured Total Least Squares-based Kalman Filter, Automatic, 56(3), pp.275-280,2016.
[2] M. Ali and A. Alireza, Robust Adaptive Unscented Kalman Filter for Bearings-Only Tracking in Three-Dimensional Case, Applied Ocean Research, 87, pp.223-232, 2019.
[3] R. K. He, S.H. Chen, H. Wu, H. Xu, K Chen, and J. Liu, Adaptive Covariance Feedback Cubature Kalman Filtering for Continuous-Discrete Bearings-Only Tracking System, IEEE access, 7, pp.2686-2694, 2019.
[4] Z. J. Yang, Application of Cubature Kalman Filter Algorithm in SINS/ BDS Integrated Navigation. Kunming University of technology, 2018.
[5] L. Ge. Research on Cubature Kalman Filter Algorithm and Its Application in Navigation. Harbin University of engineering, 2013.
[6] R. Abdolrahman, S. Behrouz and Z. Jafar, Novel Hybrid Robust Fractional Interpolatory Cubature Kalman Filters. Journal of the Franklin Institute, 357(1), 704-725, 2020.
[7] R. Rahman, M. Alanyali, and V. Saligrama, Distributed Tracking in Multi-Hop Sensor Networks with Communication Delays, IEEE Transaction on Signal Processing, 50(9), pp. 4656–4668, 2007.
[8] X. Z. Bai, L.T. Dong, L.J. Ge, H. X. Xu, J. C. Zhang and J. Yan. Robust Localization of Mobile
Robot in Industrial Environments with Non-Line-Of-Sight Situation. IEEE Access, 8, pp.22537-22545, 2020.

[9] Chen, L. Yu, W. A. Zhang, and A. Liu, Robust Information Fusion Estimator for Multiple Delay-Tolerant Sensors with Different Failure Rates, IEEE Transaction on Circuits and System, 60(2), pp. 401–414, 2013.

[10] L.J. Ge, W.L. Liao, S. X. Wang, B.J. Birgitte and R. P. Jayakrishnan Modeling Daily Load Profiles of Distribution Network for Scenario Generation using Flow-based Generative Network, IEEE Access, 2020.

[11] J. Li, H. Dong, F. Han, N. Hou, and X. Li, Filter Design, Fault Estimation and Reliable Control for Networked Time-Varying Systems, System Science and Control Engineer An Open Access Journal, 5(1), pp. 331–341, 2017.

[12] X. Z. Bai, L. Liu, M. Cao, J. Panneerselvam, Q. Sun, and H. Wang, Collaborative Actuation of Wireless Sensor and Actuator Networks for the Agriculture Industry, IEEE Access, 5, 13286-13296, 2017.

[13] H. Tan, B. Shen and Q. Li, Event-Triggered Extended Kalman Filter for Nonlinear Multi-Rate Systems, Chinese Control Conference, 396-401, 2017.

[14] S. Das and J. Moura, Consensus Innovations Distributed Kalman Filter with Optimized Gains, IEEE Transaction on Signal Processing, 65(2), 467–481, 2017.

[15] X. Z. Bai, Z. D. Wang, L. Sheng, Z. Wang, Reliable Data Fusion of Hierarchical Wireless Sensor Networks with Asynchronous Measurement for Greenhouse Monitoring, IEEE Transactions on Control Systems Technology, 27(3), 1036–1046, 2018.

[16] H. Geng, Z. D. Wang, Y. Liang, Y. Cheng, and F. E. Alsaadi, Tobit Kalman Filter with Fading Measurements, Signal Processing, 140, pp. 60–68, 2017.