Research on underwater target tracking based on Gaussian Hermitian Kalman Particle filter algorithm

Chengzhen Zhang, Yang Yang, Yuanming Ding

College of Information Engineering, Dalian University, Dalian 116622, China.
dingyuanming@dlu.edu.cn

Abstract. The underwater target tracking environment usually has the characteristics of strong nonlinearity and non-Gaussian. The target tracking problem usually uses the nonlinear filtering algorithm, which combines the nonlinear measurement model with the linear system dynamics. The primary goal of target tracking is to extract accurate information about the real-time state of the target from the noise nonlinear observation obtained by the sensor. A new Gaussian Hermitt Kalman particle filter algorithm (GHKF-PF) is used to improve the tracking accuracy of the particle filter algorithm (PF). GHKF-PF uses the GHKF sampling method to sample particles from the posterior distribution of the target, calculates the mean and covariance for each particle, and uses the mean and variance to guide sampling. In the simulation experiment, the uniform linear motion model of underwater 3D target in the environment of Gaussian mixture noise (GMN) is established, and the application of GHKF-PF algorithm in 3D space is realized, and the accuracy is higher than that of PF. In order to further verify the effectiveness of the algorithm, a six-dimensional uniform acceleration motion model is established and compared with the extended particle filter (EPF) algorithm and the unscented particle filter (UPF) algorithm. The simulation results show that the performance of the GHKF-PF algorithm is similar to that of the UPF algorithm and better than the EPF algorithm.

1. Introduction

In the research of target tracking based on autonomous underwater vehicle, the problems of observability, uncertainty and nonlinearity are mainly solved by nonlinear filtering algorithms, such as extended Kalman filter[1], Kalman filter with interpolated sigma points[3], Monte Carlo (particle filter)[4], cubic Kalman filter[5] and so on.

In order to improve the location performance, a variety of target tracking filtering algorithms are applied. An interactive MCMC particle filter method is proposed in reference[7]. The direct interactive MCMC sampling method is used to sample particles from the target posterior distribution, which avoids the lack of samples and enhances the robustness of the algorithm. In addition, the interactive MCMC particle filter algorithm propagates according to the historical information of each particle and the information of other particles, which speeds up the convergence speed of MCMC, and only considers the tracking in two-dimensional state. In reference[8], underwater target tracking based on a multistatic sonar network composed of passive sonobuoy and active PING is proposed, and a robust target tracking algorithm is proposed. The Gaussian mixture model is used to fuse multiple TDOA measurements to estimate the target position, and the effective tracking is achieved, but the target angle is not discussed. A maneuvering target tracking algorithm based on Gaussian MonteCarlo particle filter is proposed in reference[9]. The algorithm uses Gaussian and MonteCarlo methods to improve the process of particle resampling extraction to improve the efficiency and performance of
particle extraction, only considering the case of Gaussian white noise. In order to improve the estimation accuracy of nonlinear dynamic systems, a new volume Kalman filter based on spherical simplex Gauss-Laguerre quadrature is proposed in reference[10]. The nonlinear Gaussian weighted integral is approximately calculated by spherical simplex method and arbitrary order Gauss-Laguerre quadrature method, but the performance in the case of colored noise is not discussed.

In this paper, based on the Gaussian Hermit Kalman filter algorithm, the principle of the Gaussian Hermitt Kalman particle filter algorithm is analyzed, and the three-dimensional target uniform motion model is established. Compared with the particle filter (PF), the two-dimensional target uniform acceleration model is established, and compared with the extended particle filter algorithm (EPF) and the unscented particle filter (UPF) algorithm to verify the effectiveness of the algorithm.

2. Algorithm description

2.1. Gaussian Hermitt Kalman particle filter algorithm
Although the Gauss Hermite quadrature method appeared in mathematical literature more than 50 years ago, it has only recently been used in signal processing by Ito[11-12] et al. The core of GHKF is the Gauss-Hermite quadrature rule, which makes it possible to approximate the difficult integrals encountered in nonlinear Bayesian filtering problems. Generally speaking, the quadrature rule approximately calculates the complex integral by rewriting the complex integral as the product of the function of the non-negative weight and the quadrature point. The unknown probability density function is approximated to a Gaussian function by using Gauss-Hermite integral points and their special weights.

By assuming that the given posterior density function has state mean \( f_m \) and covariance \( f_p \) at time \( k = 1 \ldots \tau \), the \( g \)-th Gauss Kalman filter is solved.

\[
g(y_{k-1}|z_{k-1}) = N(f_m_{k-1}|f_p_{k-1})
\]  

(1)

2.2. GHKF-PF algorithm flow
(1) Prediction stage
Calculate the root \( x_j \) of the Hermite polynomial \( H_g(x) \) when \( j = 1 \ldots g \).
Find a specific weight \( \alpha_j \)

\[
\alpha_j = \frac{2^{g-1}j!}{g!(a_j)^2}
\]  

(2)

Apply the product rule to extend the point to a set of \( m \)-dimensional \( g^m \) points \( \psi_j \), where \( j = 1 \ldots g^m \), with a given weight.
Calculate the volume point. The square root of the matrix is set to the lower triangular Joeles gene.

\[
y^*_{j,k|k-1} = \sqrt{f_p_{k-1}} \psi_j + f_m_{k-1}
\]  

(3)

By using the dynamic model function \( Y \), the estimated volume point is as follows.

\[
y^*_{j,k|k-1} = \Phi(y^*_{j,k-1|k-1})
\]  

(4)

The predicted average value of the state function is evaluated as.

\[
f_m_{k|k-1} = \sum_{j=1}^{g^m} \alpha_j y^*_{j,k|k-1}
\]  

(5)

The covariance estimates of prediction errors are as follows.

\[
f_p_{k|k-1} = \sum_{j=1}^{g^m} \alpha_j y^*_{j,k|k-1} y^*_{j,k|k-1}^T - f m_{k|k-1} f_m_{k|k-1} + \omega_{k-1}
\]  

(6)

(2) Update phase
To obtain the \( g^m \) volume points and their corresponding weights, refer to step 3 of the prediction phase.
Calculate the volume point to.

\[
y^*_{j,k|k-1} = \sqrt{f_p_{k|k-1}} \alpha_j + f_m_{k|k-1}
\]  

(7)

By using the measurement model function, the volume point is estimated as.

\[
Z^*_{j,k|k-1} = H(y^*_{j,k|k-1})
\]  

(8)
The predicted measurements are evaluated as follows.

\[
\hat{z}_{k|k-1} = \sum_{j=1}^{m} \alpha_j Z_{j,k|k-1} \tag{9}
\]

The innovation covariance matrix is estimated to be.

\[
\Gamma_k|k-1 = \sum_{j=1}^{m} \alpha_j Z_{j,k|k-1} Z_{j,k|k-1}^T - \hat{z}_{k|k-1}\hat{z}_{k|k-1}^T + A_k \tag{10}
\]

Find the cross covariance matrix as follows.

\[
f_{p_{xz,k|k-1}} = \sum_{j=1}^{m} \alpha_j Y_{j,k|k-1} Z_{j,k|k-1}^T - f_{m_{k|k-1}} \hat{z}_{k|k-1} \tag{11}\]

The Kalman gain term, the mean value of the filtering state and the covariance of the filtered state are calculated to be.

\[
K_k = f_{p_{xz,k|k-1}} \Gamma_k|k-1^{-1} \tag{12}
\]

\[
f_{m_{k|k}} = m_{k|k-1} + K_k (z_k - \hat{z}_{k|k-1}) \tag{13}
\]

\[
f_{p_{k|k}} = f_{p_{k|k-1}} - K_k f_{p_{xz,k|k-1}} K_k^T \tag{14}
\]

The core here is to calculate the filter state mean and covariance. To facilitate the expression of the following formula, make \(X_i(t) = m_{k|k}, \hat{p}_k^{(i)} = f_{p_{k|k}}\). (3) Sampling estimation phase.

The i-th particle update.

\[
\tilde{X}_k^{(i)} = q \left( X_k^{(i)} \left| X_{0:k-1}, Z_{1:k} \right. \right) = \mathcal{N}(\hat{X}_k^{(i)}, \hat{p}_k^{(i)}) \tag{15}
\]

\[
\tilde{X}_k^{(i)} = (X_k^{(i)}, \hat{X}_k^{(i)}) \tag{16}
\]

\[
\tilde{p}_k^{(i)} = (P_k^{(i)}, \hat{p}_k^{(i)}) \tag{17}
\]

For \(i=1:N\), recalculate the weight for each particle.

\[
\omega_k^{(i)} \propto \frac{p(z_k|\tilde{X}_k^{(i)}, X_{0:k-1})}{q(\tilde{X}_k^{(i)}|X_{0:k-1}, Z_{1:k})} \tag{18}
\]

For \(i=1:N\), normalized weight.

Selection phase (resample).

The resampling algorithm is used to copy and eliminate the particle set \(\tilde{X}_{0:k}^{(i)}\) according to the size of the normalized weight \(\tilde{\omega}_k^{(i)}(X_{0:k}^{(i)})\).

For \(i=1:N\), reset the weight \(\omega_k^{(i)} = \tilde{\omega}_k^{(i)} = \frac{1}{N}\).

Output.

3. Mathematical modeling

3.1. Target uniform motion model

In the Cartesian coordinate system, considering that AUV is initially in the position of \(O(r_x, r_y, r_z)\), and assuming that the target motion is approximately linear, the velocity is kept constant. The state vector of the target at time \(t\) is expressed as.

\[
S_o(t) = [r_{x_0}(t), v_{x_0}(t), r_{y_0}(t), v_{y_0}(t), r_{z_0}(t), v_{z_0}(t)]^T \tag{19}
\]

Where \(r_{x_0}(t), v_{x_0}(t)\) represents the position and speed of the target in the \(x\)-axis direction, and the \(y\)-axis is the same as the \(z\)-axis.

Calculate the relative state vector of the next time based on the current time state vector, as shown in the formula.

\[
S_o(t+1) = \Phi S_o(t) + \Gamma W(t) \tag{20}
\]

Where \(\Phi\) is the state transition matrix calculated according to the formula.
\[ W(k) \text{ is the process noise, and } \Gamma \text{ is the process noise driving matrix.} \]

\[
\Gamma = \begin{bmatrix}
\frac{T^2}{2} & 0 & 0 \\
T & 0 & 0 \\
0 & \frac{T^2}{2} & 0 \\
0 & T & 0 \\
0 & 0 & \frac{T^2}{2} \\
0 & 0 & T
\end{bmatrix}
\]

(22)

The covariance matrix of process noise is calculated according to the formula.

\[
Q(t) = \sigma^2 \begin{bmatrix}
\frac{T^4}{2} & \frac{T^3}{2} & 0 & 0 & 0 & 0 \\
\frac{T^3}{2} & T^2 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{T^4}{4} & \frac{T^3}{2} & 0 & 0 \\
0 & 0 & 0 & \frac{T^4}{4} & T^2 & 0 \\
0 & 0 & 0 & 0 & \frac{T^3}{2} & T^2
\end{bmatrix}
\]

(23)

Where \( \sigma^2 \) represents the variance in the process noise. \( \sigma^2 \) is an adjustable parameter, and \( \sigma^2 \ll 1 \).

The AUV observation equation for this application has only distance, and the observation equation is expressed as.

\[
Z(t) = H(X(t)) + V(t - 1)
\]

(24)

\[ V(t) \text{ is the measurement noise, and it is assumed to obey the Gaussian mixture noise distribution with } \sigma_v^2 \text{ variance.} \]

The Jacobian matrix of the observation equation is given.

\[
\begin{bmatrix}
\frac{\partial Z(k)}{\partial X(k)} &=& \begin{bmatrix}
\frac{\partial Z(k)}{\partial x(k)} & \frac{\partial Z(k)}{\partial x(k)} & \frac{\partial Z(k)}{\partial y(k)} & \frac{\partial Z(k)}{\partial y(k)} & \frac{\partial Z(k)}{\partial z(k)} & \frac{\partial Z(k)}{\partial z(k)} \\
\end{bmatrix}
\end{bmatrix}
\]

\( x(k) - x_0 \)

\( y(k) - y_0 \)

\( z(k) - z_0 \)

\[
= \begin{bmatrix}
\frac{(x(k) - x_0)^2 + (y(k) - y_0)^2 + (z(k) - z_0)^2}{\sqrt{(x(k) - x_0)^2 + (y(k) - y_0)^2 + (z(k) - z_0)^2}} \\
0, \frac{(x(k) - x_0)^2 + (y(k) - y_0)^2 + (z(k) - z_0)^2}{\sqrt{(x(k) - x_0)^2 + (y(k) - y_0)^2 + (z(k) - z_0)^2}} \\
0, 0, \frac{(x(k) - x_0)^2 + (y(k) - y_0)^2 + (z(k) - z_0)^2}{\sqrt{(x(k) - x_0)^2 + (y(k) - y_0)^2 + (z(k) - z_0)^2}}
\end{bmatrix}
\]

(26)

3.2. Target uniformly accelerated motion model

A uniformly accelerated motion model is established, and a target M moving in a two-dimensional plane \( x - y \) is considered. Its position, velocity and acceleration at a certain time can be expressed by the following formula.

\[
X(k) = [x_k, y_k, \dot{x}_k, \dot{y}_k, \ddot{x}_k, \ddot{y}_k]^T
\]

(27)
Suppose that M moves approximately uniformly in a straight line in the horizontal direction (x-axis direction) and vertical direction (y-axis direction). The motion in both directions has additive system noise \( W(\mathbf{k}) \), and the motion equation of state of the target in Cartesian coordinate system is.

\[
X(k + 1) = \Phi X(k) + W(k)
\]  
(28)

\[
\Phi = \begin{bmatrix}
1 & 0 & T & 0 & T^2/2 & 0 \\
0 & 1 & 0 & T & 0 & T^2/2 \\
0 & 0 & 1 & 0 & T & 0 \\
0 & 0 & 0 & 1 & 0 & T \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]  
(29)

Assuming the AUV whose coordinate position is \((x_0, y_0)\) to track the target M, the distance \( r_\ell \) between AUV and target M and the angle of the target relative to AUV can be obtained. In the actual measurement of \( \phi_\ell \), AUV has additive noise \( V(\mathbf{k}) \). In the coordinate system centered on AUV, the observation equation is as follows.

\[
Z(k) = h(X(k)) + V(k) = \begin{bmatrix}
\sqrt{(x(k) - x_0)^2 + (y(k) - y_0)^2 + V_r(k)} \\
\arctan \left( \frac{y(k) - y_0}{x(k) - x_0} \right) + V_\phi(k)
\end{bmatrix}
\]  
(30)

Suppose that the system noise \( W(k) \) has a covariance matrix \( Q_k \), and \( V(k) \) has a covariance matrix \( R_k \), respectively as follows.

\[
Q_k = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.1^2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.1^2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.01^2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.01^2
\end{bmatrix}
\]  
(31)

\[
R_k = \begin{bmatrix}
5 & 0 \\
0 & 0.01^2
\end{bmatrix}
\]  
(32)

### 3.3. Gaussian mixture noise model

Gaussian white noise is a simple random process, which is an idealized random process composed of a series of unrelated random variables. The density of Gaussian mixture noise has different variance, different samples are different, and the data with the condition of variance is normal. Therefore, GMN is generated by adding Gaussian noise to Gaussian noise with variable variance.

Use ‘randn’ in MATLAB to design, and the designed GMN is shown below.

\[
\text{GMN}_\text{noise}(t) = \text{Gaussian}_\text{noise}(t) + 0.2 \ast \text{Gaussian}_\text{noise}(t - 1) \\
+ 0.7 \ast \text{Gaussian}_\text{noise}(t - 1) + 1.2 \ast \text{Gaussian}_\text{noise}(t - 1)
\]  
(33)

In the above formula, \((0.2\ast\text{randn})\), \((0.7\ast\text{randn})\) and \((1.2\ast\text{randn})\) generate Gaussian noise with different variances respectively. Add these noises to form a GMN of random variance.

Figure 1 shows the simulated GMN histogram, 1800 samples are simulated, and the distribution is approximately Gaussian. The difference from Gaussian white noise is that this sample is a sample with an amplitude greater than 1 degree. For about 400 random samples, noise with a higher amplitude is applied. The results of multiple simulations show that the average value of the noise amplitude is not zero, and the variance does not exceed 6.

Figure 2 shows the frequency spectrum of GMN. It can be seen that GMN is a kind of colored noise, which is characterized by that as the frequency increases, the spectrum gradually weakens at \( k = 300 \), while the spectrum of Gaussian white noise is uniformly distributed.
4. Simulation results

The GHKF-PF algorithm is implemented in the MATLAB PC environment, and compared with the traditional particle filter algorithm to evaluate its underwater target tracking effect.

4.1. Uniform moving target tracking

Assuming that the measurement results per second are continuously available, the AUV is located at (200,300,100)m, the target is at(−100,200,150)m, and it is approximately moving in a straight line at a constant speed. Its initial speed is (2,5,5)m/s. The process noise variance $\sigma_2 = 10^{-3}$ and remains unchanged. The observation noise $\sigma_2^2$ of the AUV is determined according to the variance of the Gaussian mixture noise of each experiment, and the amplitude does not exceed 6. And set the sampling interval $T$ of the sample to 1s, the simulation time is 60 seconds, the number of particles $N_p=50$, the algorithm does 50 Monte Carlo simulations, and obtains the tracking image, the position error and the RMSE of the 50 simulations under a certain experiment Graph.

Figure 3 shows the tracking curve of the two algorithms. In the early stage of tracking, both algorithms can keep good tracking. With the passage of time, the PF algorithm becomes unstable, and GHKF-PF keeps a good tracking curve. The main reason is that PF is greatly affected by non-Gaussian noise. Figure 4 shows the position error curve, and the error of PF tends to increase with time.

Figure 5 shows the RMSE values of 50 Monte Carlo experiments. The RMSE distribution of PF is uneven, indicating that the stability of the algorithm is not strong, and GHKF-PF is more robust than
PF. In 50 Monte Carlo simulations, record the RMSE values of the two algorithm simulations, sum and average them, and the average error of the particle filter is 7.5488, and the error of GHKF-PF is 6.8379. The statistics of many experiments show that the accuracy of GHKF-PF algorithm is better than that of PF algorithm.

![Fig.5 RMSE error under each experiment](image)

4.2. Uniformly accelerated moving target tracking

In order to further demonstrate the effectiveness of GHKF-PF algorithm, this algorithm is compared with extended particle filter and unscented particle filter.

In the simulation, set the sampling time \( t = 0.5s \), sample 60 times, AUV at \((0,1000)\), initial target state.

\[
X(0) = [1000,5000,10,50,1, -3]^T
\]  

(34)

Initialize covariance.

\[
P_0 = \begin{bmatrix}
10 & 0 & 0 & 0 & 0 & 0 \\
0 & 10 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.1 \\
0 & 0 & 0 & 0 & 0 & 0.1
\end{bmatrix}
\]  

(35)

After 50 Monte Carlo simulations, the tracking trajectories of EPF, UPF and GHKF-PF generated in one experiment are shown in figure 6, the tracking position error is shown in figure 7, and the RMSE value of each experiment is shown in figure 8.

![Fig.6 Tracking curve of uniform acceleration motion](image)  ![Fig.7 Error curve](image)
In figure 6, all three algorithms can track the target trajectory well. Figure 7 shows the specific difference of each sampling and tracking. It can be seen that the measurement error of EPF increases with the increase of sampling times, and the errors of HKF-PF and UPF are the same and remain at a low level. The RMSE value of figure 8 further illustrates this point and is more robust. The RMSE value of EPF is uneven, which reflects the lack of stability of the algorithm.

Fig.8 RMSE value under each experiment

5. Conclusion
In this paper, an improved Gaussian Hermitt Kalman particle filter GHKF-PF algorithm is analyzed, and the underwater 3D target tracking model under Gaussian mixture noise (GMN) is established, and the effectiveness of the algorithm is verified by simulation. Furthermore, under the six-dimensional uniform acceleration motion tracking model, the target tracking performance of GHKF-PF, EPF and UPF algorithms are analyzed and compared. The performance of GHKF-PF algorithm is similar to that of UPF, robust, and better than EPF.

Acknowledgments
This work was funded by the National Natural Science Foundation of China. (61540024).

References
[1] MILLER A B, MILLER B M. Underwater target tracking using bearing-only measurements [J]. Journal of Communications Technology and Electronics, 2018, 63(6): 643-649.
[2] XIN G, XIAO Y, YOU H. Bearings-only underwater track fusion solutions with feedback information [C]/Proceedings 7th International Conference on Signal Processing(ICSP04), 2004, 3: 2449-2452.
[3] GUPTA R, KUMAR A, KAR I N, et al. Bearings-only tracking of non-maneuvering target with missing bearings data [C]/ 2015 IEEE Underwater Technology (UT), 2015: 1-7.
[4] FEI Z, XING-PENG Z, XIAO-HUI C, et al. Particle filter for underwater bearings-only passive target tracking [C]/ 2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application, 2008, 2: 847-851.
[5] Guo Chengjun, Zhang Xinchun. Cubature kalman filters: derivation and extension [J]. Chinese Physics B, 2013, 22(12): 128401(in Chinese).
[6] LEONG P H, ARULAMPALAM S, LAMAHEWA T A, et al. A Gaussian-sum based cubature kalman filter for bearings-only tracking [J]. IEEE Transactions on Aerospace and Electronic Systems, 2013, 49(2): 1161-1176.
[7] JING L, VADAKKEPAT P. Interacting MCMC particle filter for tracking maneuvering target [J]. Digital Signal Processing, 2010, 20(2): 561-574.
[8] SHIN H, KU B, NELSON J K, et al. Robust target tracking with multi-static sensors under insufficient TDOA information [J]. Sensors, 2018, 18(5): 1481.

[9] Liu Hongqing, Liu Yan, Wu Junliang. Maneuvering target tracking algorithm based on gauss monte carlo particle filter [J]. Control Engineering, 2018, 25(09): 1754-1759 (in Chinese).

[10] CAO T, GAO H, SUN C, et al. Application of improved simplex quadrature cubature kalman filter in nonlinear dynamic system [J]. Mathematical Problems in Engineering, 2020, (5):1-13.

[11] KRYLOV V I, STROUD A H. Approximate calculation of integrals [M]// Approximate calculation of integrals /. Dover Publications, 2005.

[12] ITO K, XIONG K. Gaussian filters for nonlinear filtering problems [J]. IEEE transactions on automatic control, 2000, 45(5): 910-927.