Research on Performance of Terrain Matching Algorithm for Underwater Autonomous Vehicle Based on Particle Filter

ShenJian¹, Xiong Lu² and Ba Ning³

¹ Post-doctoral program on control science and engineering, Naval University of Engineering, Wuhan 430033, China
² Department of Radar system, OrdnanceNCO Academy of Army engineering university, Wuhan 430075, China

Abstract. AUV is a powerful tool for exploration and development of marine resources. Underwater terrain matching navigation is an emerging passive autonomous navigation method, which improves the long-term and high-precision navigation capability of AUV. The underwater terrain matching algorithm based on particle filter (PF) use a large number of particle sampling to approximate the system state, which can avoid the Taylor expansion truncation error caused by the terrain linearization method, and has good matching precision and terrain adaptability. However, due to the strong nonlinearity and the uneven distribution of the underwater terrain features, it is difficult to achieve the full-range terrain matching optimize with a single configuration parameter of PF underwater terrain matching algorithm. For this reason, this paper has comprehensively studied the PF terrain matching algorithm (referred to as PF matching algorithm). The principle, accuracy and implementation steps of PF algorithm are analyzed. With the UKF algorithm as a reference, the advantages and disadvantages of PF-based algorithm are studied through computer simulation of terrain matching navigation. The matching precision and terrain adaptability of PF-based algorithm has been tested. From the simulation results, it is concluded that the particle diversity decline is one of the main factors affecting the matching performance, which provides a basis for algorithm performance improvement and engineering adaptability improvement.

1. Introduction
The navigation system is one of the most basic components of the underwater autonomous vehicle(AUV). It provides real-time position, velocity and attitude information for the AUV, which is the important Information input for AUV’s autonomous control and navigation [1]. Underwater terrain matching navigation is a new type of passive, autonomous navigation method. The AUV’s current position can be estimated by comparing the real-time measured terrain data with the underwater terrain database. From the perspective of estimation theory, the terrain matching algorithm essentially uses the observable measurement data to estimate the AUV’s state variable by the estimation method [2]. Therefore, the recursive filtering is considered to be one of the effective ways to solve the underwater terrain matching problems.

In the 1970s, the terrain matching algorithm based on extended Kalmanfilter(EKF) has been successfully applied on the missiles. As the first-order linear truncation of Taylor expansion is used for extended Kalman filter’s (EKF) terrain linearization[3], the state variables estimated precision of EKF...
algorithm is low. In addition, the Jacobian matrix solution of the system state equation in engineering application is usually complicated, which affects the system response speed to some extent[4].

With the continuous development of filtering technology, PF nonlinear filtering method is used as the filtering kernel of the terrain matching algorithm, which improves the filtering precision of the matching algorithm. However, due to the strong nonlinearity of the underwater terrain and the randomness of the terrain feature distribution, it is difficult to achieve the ideal matching accuracy and robust effect by using the single PF based terrain matching algorithm in a region with varied terrain[5,6]. For this reason, it is necessary to optimize the PF parameters and algorithm structure in order to adapt the different terrain features. This paper fully analyze the performance of PF based terrain matching algorithm, firstly, the system state approximation principle of PF algorithm is introduced, the precision and characteristics of the PF algorithm are analyzed compared with the UKF algorithm. Secondly, the simulation of univariate nonstationary growth model (UNGM) is used to study the two algorithms’ estimation performance. Thirdly, the AUV underwater terrain matching framework based on PF algorithm is given. The underwater terrain matching model is established and the underwater terrain matching simulation has carried out by using the two matching algorithms. Finally, the positioning effect of the terrain matching algorithms are studied to provide reference for the AUV navigation under different application conditions.

2. PF algorithm analysis

2.1. Principle of PF algorithm

Particle filtering is based on Bayesian estimation, Markov process and Monte Carlo theory. The core idea of PF is to simulate the probability distribution of system parameter by large random sampling from state sample set [7]:

\[ p(x_{ik} | z_{ik}) \approx \sum_{i=1}^{N} \omega_i \delta(x_{ik} - x_{0ik}) \]  

(1)

The \( \{x_{0ik}, \omega_i\}_{i=1}^{N} \) is defined as the particle set of the posteriori probability density function \( p(x_{0ik} | z_{ik}) \), the \( \{x_{0ik}, i=0, \ldots, N_i\} \) corresponding to the weight coefficient set \( \{\omega_i, i=0, \ldots, N_i\} \) is representing the particle set of system states, \( x_{0ik} = \{x_i, j=0, \ldots, k\} \) representing the system state set from the initial moment to the k-th moment, \( z_{ik} = \{z_j, j=0, \ldots, k\} \) representing all observations from the 1-th moment to the k-th moment, \( p(x_{0ik} | z_{ik}) \) representing the probability density under the observations sequence.

It can be seen from equation (1) that the more the number of particles, the better the probability approximating by the particle set \( \{x_{0ik}, \omega_i\}_{i=1}^{N} \). The posterior probability density of k-th moment can be approximated by using the particle set \( \{x'_i, \omega'_i\}_{i=1}^{N} \) at the corresponding moment, that is, by calculating the particle set at each moment, the recursive law of the corresponding system state can be obtained.

2.2. Comparison of PF algorithm and UKF algorithm

2.2.1. UKF algorithm. UKF is a nonlinear Kalman filtering framework method based on the Unscented transformation theory. The core idea is to achieve nonlinear approximation of the state distribution by deterministic sampling. Compared with the EKF, UKF’s nonlinear estimation precision can reach above 2nd-order Taylor expansion truncation precision, and the approximate calculation is a Non-Jacobian matrix operation process, which is suitable for engineering realization[4,8].

The sigma point sampling of UKF is the basics for nonlinear state approximation, Commonly used sampling methods is:

- symmetric distributed sampling, the representative algorithm is the symmetric set sampling (Standard UKF)[8];
• hyper-spherical sampling, the representative algorithm is volume Kalman filter (Cubature Kalman Filter, CKF);
• Gaussian distributed volume sampling, the representative algorithm is Gauss-Hermite Kalman Filter (GHKF) [9,10].

If the system transfer equation is developed by Taylor expansion representation, the UT transformation can ensure the correct recursive of system state not higher than the 2nd order. When the distribution of Independent variables is symmetrical (symmetric distributed sampling), the UT transformation can achieve the precision of 3rd order. When the independent variable satisfies the Gauss distribution (Gaussian distributed volume sampling), for the one-dimensional case, the sampling method can achieve 4th-order precision, while for multi-dimensional variables, the UT transform can only guarantee the 4th-order axis directional precision.

By definition, the minimum mean square error can be classified as criterion of the norm minimum, which is not robust in statistics. Therefore, when the assumptions are deviated from the actual parameters, the estimation results will be greatly affected [4]. Based on the above reasons, the UKF-based underwater terrain matching algorithm will also face the lack of robustness.

2.2.2. PF algorithm. From the sampling principle of particle filtering, if the particles number is sufficient, the particle set can approximate the System state variables with arbitrary precision. Unfortunately, with the increasing of recursion time k, most particle weights are concentrating on a few particles, while the other particles’ weights are too small to effective. This phenomenon is called the particle degradation which seriously affects the algorithm computational efficiency. Therefore Gordon.N.J proposed the resampling method, which removes the small weight particles and preserves the larger weight particles. This sampling method can avoid the repeated calculation of small weight particles in recursion, which ensures the computational efficiency[7].

Compared with the N=2n+1(n is the dimension of system state) sampling points of UKF algorithm, the PF system need all particles to participate the states/variance estimation in each recursive step, the calculation amount is much larger than the UKF algorithm. Therefore, reasonable optimization must be made between the precision of the algorithm and the calculation speed according to the actual application. In addition, as a fixed component of the particle filter framework, resampling can effectively solve the particle degradation problem, however, since high-weight particles are copied many times in the iterative process, a large number of particles are clustered the high-weight state region. It reduces the diversity of particle types, affects the state approximation ability of particle set, reduces the accuracy and robustness of the algorithm.

3. Univariate nonstationary growth model simulation
In this section, the univariate nonstationary growth model (UNGM) [4,5,7] is used to test the performance of the UKF and PF. The UNGM has the characteristics of high nonlinearity and bimodal state distribution, so it is suitable for verifying the nonlinear estimation ability of filter algorithm. The UNGM discrete dynamic space model can be expressed as:

\[ x_k = 0.5x_{k-1} + 25 \frac{x_{k-1}^2}{1+x_{k-1}^2} + 8\cos(1.2(k-1)) + u_{k-1} \]  \[ y_k = \frac{x_k^2}{20} + v_k \quad k = 1, 2, \ldots, N \] 

Where, \( u_k \sim N(0, \sigma_u^2) \) and \( v_k \sim N(0, \sigma_v^2) \) respectively represent the system noise and observed noise which satisfy the Gaussian distribution. Taking \( \sigma_u^2 = \sigma_v^2 = 1 \). In order to increase the contrast, the UKF algorithm is used as the comparison algorithm in the simulation. The sampling scale factor of the
UKF is $\alpha = 0.5$, $\beta = 2$, $\kappa = 1$, and the PF particle number $N_i=500$, the simulation time is $t=100$, the Monte Carlo simulation time is $N=500$. The Mean Square Error (MSE) is define as:

$$
\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})^2, \quad N = 500
$$

The simulation results are shown in Figure 1.

(a) Estimated error of UKF2 and PF
(b) Estimated MSE and $3\sigma$ confidence interval of UKF
(c) Estimated MSE and $3\sigma$ confidence interval of PF

**Figure 1.** Estimated error and Estimated MSE of UKF and PF

In the simulation result, the PF shows a better estimate ability than that of the UKF. This because the state distribution approximates of UKF is just first/second order, while PF achieves an overall approximation of the conditional probability distribution through a large number of particles sampling, and thus has higher precision than that of UKF. Comparing Figure 1(a) and Figure 1(b), the estimation error of PF algorithm almost falls within the $3\sigma$ confidence interval, which reflects a high estimation result confidence of algorithm. In addition, the $MSE_{\text{UKF}}=19.1991$, $MSE_{\text{PF}}=9.3434$ are also prove a better estimation accuracy of the PF algorithm. However, the average Monte Carlo simulation time of UKF is $t_{\text{UKF}}=10.32$ms, while the average simulation time of PF is $t_{\text{PF}}=48.16$ms (QuadCore Inter Core i7 7700HQ,3800MHz, None GPU computing mode ). It indicates that better estimation accuracy of particle filter is at the expense of calculation amount increasing.

4. **PF-based underwater terrain matching algorithm**

4.1. **PF-based underwater terrain matching algorithm**

The AUV underwater navigation mode includes fixed depth navigation and terrain following navigation. Since the pitching motion of the AUV easily causes the depth measurement error, the AUV usually uses fixed-depth navigation in the matching process. To this end, different terrain matching algorithms can be studied by the two-dimensional underwater terrain matching model [6]:

$$
\begin{align*}
X_{\text{t+1}} &= X_t + \Delta s + \omega \\
\zeta &= h(X_t) + \zeta
\end{align*}
$$

Where, $X_t = (x, y)$ is the current position of the AUV, $\Delta s$ indicating the distance between the two matching positions, the system noise $\omega$ and the measured noise $\zeta$ are zero mean white Gaussian noise, $Q$ and $R$ are the variances are respectively.

4.2. **Basic framework of PF underwater terrain matching algorithm**

The PF underwater terrain matching algorithm includes four parts: Initialization, time update and measurement update, and resampling[6,7].
• Initialization: Passing the navigation information at the AUV starting matching position to the particle set, so that the AUV navigation information can be continuously corrected in the subsequent recursion;
• Time update: PF predicts the position and its variance of the next matching time through the navigation equation and the state particle set;
• Measurement update: The predicted position and its variance are adjusted according to the difference between the measured depth and the estimated depth.
• Resampling: Eliminate small weighted particles i and copy large weighted particles n the particle set to obtain a new posterior particle set.

4.3. Algorithm implementation

• Particles Initialization: Particles initialization is performed using the AUV navigation initial position \( \hat{X}_0 = (\hat{x}_0, \hat{y}_0) \) and its variance \( P_0 \). The \( \{ \chi^i_0, \omega^i_0 \}_{i=1}^N \) is the particle set of the prior probability \( P_0 \), where \( \chi^i_0 = (x^i_0, y^i_0) \) is the AUV positional information of the i-th particle. The weight of the i-th particle is \( \omega^i_0 = \frac{1}{N} \).

• Time update: The particles \( \chi^i_{k+1} \) \( i = 1, 2, \ldots N \) is updated using equation (5) to obtain the predicted particles set \( \{ \chi^i_{k+1}, \omega^i_{k+1} \}_{i=1}^N \) in time k+1.

\[
w^i_{k+1} = w^i_k \cdot p\left( z_k | x^i_k \right) = w^i_k \cdot p\left( z_k - h\left( x^i_k \right) \right) \quad i = 1, 2, \ldots N.
\]

The predicted position of AUV at time k+1 can be calculated using equation (7):

\[
\hat{X}_{k+1} \approx \sum_{i=1}^N w^i_{k+1} x^i_{k+1}
\]

• Measurement update: Let \( \chi^i_{k+1, \text{kal}} = \chi^i_{k+1, \text{kal}} \), The predicted positions of the particles \( \chi^i_{k+1, \text{kal}} \) \( i = 1, 2, \ldots N \) can be find in the underwater digital map. The depth information \( h_{k+1} \) \( i = 1, 2, \ldots N \) of these positions are obtained by terrain interpolation algorithm. Compare these predicted depths with the measured depth \( z_k \) to complete the update of particle set weights:

\[
\omega^i_{k+1, \text{kal}} = \omega^i_{k+1} \cdot p_{z_k}\left( z_k - h^i_k \right) \quad i = 1, 2, \ldots N.
\]

The AUV Posterior position and its estimate variance at time k+1 can be obtained:

\[
\hat{X}_{k+1} = \sum_{i=1}^N \omega^i_{k+1, \text{kal}} \cdot \chi^i_{k+1, \text{kal}}
\]

\[
\hat{P}_{k+1} = \sum_{i=1}^N \omega^i_{k+1, \text{kal}} \left( \chi^i_{k+1, \text{kal}} - \hat{X}_{k+1} \right) \left( \chi^i_{k+1, \text{kal}} - \hat{X}_{k+1} \right)^T
\]

• Resampling: The particle set resampling threshold is:

\[
N_{\text{eff}} = \frac{N_s}{1 + \text{Var}(\omega^i_{k+1})} \approx \frac{N}{\sum_{i=1}^N \omega^i_{k+1, \text{kal}}^2}
\]

When \( N_{\text{eff}} \leq N_s \), it indicates that the particle degradation is severe and needs to take the resampling measures. Otherwise go to time update step. By repeating the above recursive process, the AUV position can be estimated in real time, and the matching track of the AUV is obtained.

5. Underwater terrain matching simulation research
In this section, the computer simulation of underwater terrain matching navigation based on PF will be carried out in order to test the matching effect of algorithm.

The terrain matching digital map is constructed by using the depth data of a certain sea area (2500m×2500m). The map resolution is 6m, the symbols ★ and ● are the starting position and the ending position of the terrain matching routes, and the matching route of two simulations are shown in Figure 2. The terrain matching route shows that the terrain features at 0-1500m is obviously undulating, while the terrain at the 1500m-2500m gradually becomes flat.

![Terrain matching route I](image1)

![Terrain matching route II](image2)

![Depth (route II)](image3)

**Figure 2.** The two simulation routes and the depth along the route II.

For comparison, the terrain matching algorithm based on UKF is simulated and analyzed under the same navigation conditions. The PF underwater terrain matching algorithm model is shown in (5)-(11) above, and the other simulation condition is show in Table 1.

**Table 1.** Terrain matching simulation parameters setting

| Navigation parameter                      | System parameters               |
|-------------------------------------------|---------------------------------|
| Initial deviation of position: (55m,55m) | Measurement error variance: 0.2m² |
| Position deviation (east): 0.1m/s         | Measurement interval :5s        |
| Position deviation (north): 0.1m/s        | Particle number: n=1000         |
| Initial velocity error : 0.2m/s           | UKF factors: α = 0.5            |
| Initial heading error : 0.3°              | UKF factors: β = 2              |
| AUV velocity: 2m/s                        | UKF factors: κ = 1              |
| Initial variance of position: diag[(50m)²,(50m)²] |                                  |

![Positioning error of route I](image4)

![Positioning error of route II](image5)

**Figure 3.** Positioning error of terrain matching method based on PF and UKF

The simulation result under the route I condition is shown in Figure 3(a). During the 0-500m, the positioning error reducing speed of the PF-based terrain matching algorithm is faster than the UKF ones. And during the route of 0-1500m, the terrain feature along the route is sufficient, the PF-based algorithm shows a better terrain matching accuracy. The above simulation results illustrates that the state distribution approximation by a large number of particles is more accurate than the approximation of only 2-n order Taylor expansion.
However, in the 1500m-2500m, as the terrain gradually becomes flat, the positioning error of the PF-based algorithm increases faster than the UKF-based algorithm. In order to prove that the above phenomenon is not caused by the terrain distribution, the starting position and the ending position of the terrain matching routes is exchanged, the AUV is navigate from the terrain flat area into the Terrain-rich area, as shown in Figure 2 (b).

It can be seen from the figure that under the condition of matching route II, the terrain near the AUV starting point is relatively flat, and the terrain features gradually become obvious with the extension of the route. Keep the simulation parameters unchanged and simulate again. The result is shown in Figure 3(b).

Comparing the Figure 3(a) with Figure 3(b), it can be found that the matching performance of the PF-based algorithm is better than the UKF algorithm regardless of the terrain distribution, and the positioning error can be kept within 10m. However, as the matching process progresses, the positioning error of the PF algorithm gradually deteriorates. At 2000m, the positioning error is greater than the UKF matching algorithm.

The simulation results above show that the PF terrain matching algorithm has better matching accuracy and terrain adaptability than the UKF-based ones, but its good matching performance is weakened with the passage of time because of the particle diversity degradation.

6. Conclusion
This paper starts with the principle of PF algorithm, the principle, matching precision and implementation steps have been studied. the advantages and disadvantages of PF matching algorithm are deeply studied through terrain matching simulation. The simulation results show that the PF-based terrain matching algorithm has the better matching accuracy and terrain adaptability, but the particles diversity degradation is one of the main factors affecting the performance of the algorithm. In the follow-up, the research team will study the degradation of particles diversity, find an effective method to reduce the degradation rate, and improve the engineering practicability of the algorithm.

Acknowledgments
ShenJian and Xiong Lu contributed equally to the writing of this article.

References
[1] Shen Gongxun, Sun Jianfeng. Application of information fusion theory in inertial/astronomical/GPS integrated navigation system [M]. Beijing: national defense industry press, 1998: 56-81.
[2] Tian Fengmin. Research on underwater terrain navigation model solving and navigation cell selecting[D]. Harbin: Harbin engineering university.
[3] Kalman R, Bucy R. New Results in Linear Filtering and prediction Theory [J]. Journal of Basic Engineering, 1961, 82(1):35-46.
[4] Chang Guobin. Nonlinear Kalman filtering algorithm and its application in attitude estimation problems [D]. Wuhan: Naval University of Engineering, 2011.
[5] Merwe R. Sigma-point Kalman filters for probabilistic inference in dynamic state-space models [D]. Portland: Oregon Health & Sci. Univ, 2004.
[6] Andrea L. Estimation Methods for Terrain Navigation[D]. Stockholm: School of Electrical Engineering, 2006.
[7] Gordon N, Salmond D. Novel approach to non-linear and non-Gaussian Bayesian state estimation[J]. Proceedings of Institute Electric Engineering, 1993, 140(2):107-111.
[8] Karlgaard C. Huber-based unscented filtering and its application to vision-based relative navigation[J]. Radar, Sonar, Navigation, 2010, 4(5):744-745.
[9] Wu Y, Hu D. A Numerical-Integration Perspective on Gaussian Filters [J]. Transactions on Signal Processing, 2006, 54(8): 2910-2921.
[10] Ienkaran A, Simon H. Cubature Kalman Filters [J]. IEEE Transactions on automatic control, 2009, 54(6):1254-1269.