

PARTICLE SWARM OPTIMIZATION FOR UNDERWATER GRAVITY-MATCHING: APPLICATIONS IN NAVIGATION AND SIMULATION

MingDa OuYang^{1,2,3}, Haopeng Fan¹, Peng Du^{2,3*}

¹ Institute of Geospatial Information, Information Engineering University, Zhengzhou, 450001, China

² Xi'an Research Institute of Surveying and Mapping, Xi'an, 710054, China

³ Stake Key Laboratory of Geo-information Engineering, Xi'an, 710054, China

Commission III, WG III/1

KEY WORDS: Gravity matching navigation; Particle swarm optimization; Affine transformation.

ABSTRACT:

This paper developed a two-stage solution for underwater gravity matching navigation based on the particle swarm optimization algorithm and affine transformation. The first stage established a starting point, and the second stage treated the matching track gained through affine transformation as a particle at the same starting point, followed by the application of the particle swarm optimization algorithm to obtain the optimal solution. To avoid falling into a local optimal solution, a convergence factor was incorporated into the optimization process in addition to the linear decreasing weight. This was followed by the addition of a constraint on the velocity and position of the particles, which was then updated in an iterating process. Two simulated navigation tracks were employed for experiments. The results revealed that the algorithm was capable of matching actual tracks in real time. Additionally, the results were found to be consistent with those obtained from the real-world tracks, with all the locations and gravity anomaly deviations falling within a tolerable range. However, when there were too many matching track points, the algorithm efficiency declined in terms of calculation time. This entails improving the algorithm through the segmentation technique.

1. INTRODUCTION

By aligning the inertial navigation system with gravity field information, it is feasible to significantly reduce error accumulation and thereby meet the navigation requirements of long-endurance, high-precision, autonomy, and concealment of underwater vehicles [Han et al., 2016; Yang et al., 2017; Wang et al., 2019]. The matching algorithm is referred to as the core of gravity-matching navigation. The fundamental premise is to acquire the optimal approximation of the real position through certain techniques, such as sequence matching and single-point matching. The sequence matching algorithm carries out correlation analysis on the entire set of track points and performs post-hoc calculations while assuring sufficient sampling points. The algorithm often encounters a certain time delay. The ICCP [Wang et al., 2013] and TERCOM [Wei et al., 2017] algorithms are two examples of such a process. When the initial position error is substantial, the ICCP algorithm may simply fall into a local optimum, resulting in the matching error of the nearest contour point. As a result, additional techniques have been applied to enhance the local convergence and real-time performance of the ICCP algorithm, including the sliding-window technique to reduce the amount of data required to calculate the nearest point [Liu et al., 2003], the chaotic optimization algorithm [Yuan et al., 2010], the triangle constraint model [Yang et al., 2014], the backpropagation neural network [Huang et al., 2011], and the constrained contour [Liu et al., 2011]. The TERCOM algorithm considers only the translation and ignores the rotation and scale changes induced by linear error, angle accumulation error, and random error of

the inertial navigation system. Thus, the precision and reliability of the matching navigation results were reduced. There are numerous techniques available for enhancing its performance, including Kalman filters [Wei et al., 2017], ICCP algorithms [Wang et al., 2011], and particle filters [Han et al., 2016]. The single-point matching algorithm overcomes the time delay problem of the sequence matching algorithm; typical versions include the SITAN algorithm, and particle filters. The SITAN algorithm requires prior knowledge of physical parameters such as the state transition matrix, stationary white noise matrix, and weight matrix of the underwater vehicle. To obtain a range of slopes in various directions, it is required to perform random linear fitting on the local geophysical field in order to circumvent the limitations of the equation, which can only cope with linear changes. In cases where the linear error is large, the filter diverges [Wei et al., 2017]. In contrast to the SITAN algorithm, the particle filter algorithm generates random samples from the empirical conditional distribution of the state vector and adjusts the position and weight of the particles based on the observed information. Hence, it is applicable to addressing situations that are nonlinear in nature. Upon reaching a sufficient number of particles, the corrected conditional distribution converges to the real conditional distribution of the state vector [Wang et al., 2016].

Particle Swarm Optimization (PSO) is a matching navigation algorithm, that was originally developed in the context of bird predation: When birds are on the prowl for food, the coordinates of the food are unpredictably distributed. The most straightforward and most effective search strategy is to delineate an area as close as possible to the food source. Individual birds

*This work was supported in part by the Program of China National Key R&D under Grant 2020YFB0505801, in part by the Program of the National Natural Science Foundation of China under Grant 42174001, in part by the National Laboratory Fund under Grant 2021WHZZB1001.

continuously modify their positions in order to find food through information sharing and individual cognition[Grandi et al.,2012].Gravity-matching navigation is essentially an optimization problem. The particle swarm optimization algorithm and the affine transformation were used in conjunction in this study to provide a two-stage positioning solution for real-time underwater gravity matching.

2. PARTICLE SWARM OPTIMIZATION

2.1 Principle

Each particle in the PSO algorithm is defined by a position and velocity specified in a D-dimensional space (i.e., D-dimensional vectors). The particles continue to move and search in the space, impacted by other particles in the group and by themselves. The level of influence is determined by the fitness function. The optimal solution is obtained through iterative calculation while simultaneously recording both the local and global extrema. The position of the i-th particle at time t is expressed as:

$$X_i^t = (x_{i1}^t, x_{i2}^t, x_{i3}^t, \dots, x_{iD}^t)^T, x_{i1}^t \in [x_{\min,d}, x_{\max,d}] \quad (1)$$

where D is the particle dimension, $i=1,2,3,\dots,N$, N is the number of particles, and $x_{\min,d}, x_{\max,d}$ are the maximum and minimum values of the search space, respectively. The particle velocity reads as

$$V_i^t = (v_{i1}^t, v_{i2}^t, v_{i3}^t, \dots, v_{iD}^t)^T, v_{i1}^t \in [v_{\min,d}, v_{\max,d}] \quad (2)$$

where $v_{\min,d}, v_{\max,d}$ are the maximum and minimum velocities, respectively. The extremum values of the particle are:

$$P_i^t = (p_{i1}^t, p_{i2}^t, p_{i3}^t, \dots, p_{iD}^t)^T \quad (3)$$

The global extremum of the swarm is:

$$g^t = (g_1^t, g_2^t, g_3^t, \dots, g_D^t)^T \quad (4)$$

Each particle determines its motion speed, adjusts the motion trajectory, and moves towards the optimal point based on its own and the group's motion experiences. Particle position and velocity are updated using the following equation:

$$v_{id}^{t+1} = \omega v_{id}^t + c_1 r_1 (p_{id}^t - x_{id}^t) + c_2 r_2 (g_d^t - x_{id}^t) \quad (5)$$

According to Equation (5), the motion of a particle can be classified into three components: its own inertial speed, the "cognitive" speed of the local optimal particle, and the "guiding" speed of the global optimal particle[Golubovic et al.,2007], where r_1 and r_2 are random numbers within the range[0,1]. Additionally, ω , c_1 , c_2 represent the weights of the three components. More precisely, ω is the inertia weight, which is employed to mitigate the influence of the previous speed on the current speed. The larger the value of ω , the better the global search ability of PSO, and the smaller the value of ω , the better the local search ability. Furthermore, c_1 and c_2 are the acceleration and learning factors, respectively, which are used to adjust the maximum stage size of the movement towards the global optimal particle and the local optimal particle. Appropriate values for c_1 , c_2 (which are typically positive) can expedite the convergence and avoid the local

optimum. In this study, these factors are assumed as $c_1=2.25$ and $c_2=2.25$.

To prevent the particles from falling into a local optimum, the following convergence factor was introduced to constrain the particle speed using the linear decreasing inertia weight PSO algorithm.

$$\omega = \omega_{start} - (\omega_{start} - \omega_{end})t / T_{max} \quad (6)$$

$$x_{id}^{t+1} = x_{id}^t + \gamma v_{id}^{t+1} \quad (7)$$

where, T_{max} is the maximum number of iterations, ω_{start} is the initial inertia weight, and ω_{end} is the inertia weight at the maximum iteration. Experiments demonstrated that the PSO algorithm converged quickly when $\omega \in [0.4, 0.9]$, with the convergence factor, γ , as follows:

$$\gamma = 2 / \left| 2 - \psi - \sqrt{\psi^2 - 4\psi} \right|, \psi = c_1 + c_2 > 4 \quad (8)$$

2.2 Fitness Function

The fitness function is critical in directing the particle toward the optimal position. For sequence correlation matching algorithms, common operators include cross-correlation(COR),mean absolute difference (MAD), and mean square deviation (MSD). Here, COR has the worst stability and accuracy, making it unsuitable for use as a decisive indicator. MAD and MSD have comparable performance and are both superior to COR. MSD equals the square of MAD, which is essentially a magnified indicator. Therefore, MSD was utilized in this study.

$$MSD = \frac{1}{N} \sum_{i=1}^N (g_i(x, y) - \tilde{g}_i(x+tx, y+ty))^2 \quad (9)$$

In Equation (9), $g_i(x, y)$ is the measured gravity anomaly of the inertial navigation system (INS), and $\tilde{g}_i(x+tx, y+ty)$ is the gravity anomaly of the matching track point extracted from the background image. Additionally, tx and ty are the offsets in latitude and longitude of the matching point relative to the INS indicated track point (x, y) , respectively. When MSD is the smallest, the correlation is the best, and the optimization is the best at the current position.

3. MATCHING NAVIGATION SCHEME

Let the measured gravity anomaly sequence be $LS = \{LS_1, LS_2, LS_3, \dots, LS_L\}$, where $LS_i = (B_{INS}, L_{INS}, \partial g)$ is the latitude, longitude, and measured gravity anomaly output of the inertial navigation system.

The two-stage method was employed for matching navigation. The critical component in the first stage was to determine the starting point. A search area was formed by centering it on the starting point of the inertial navigation system output[17]. The particles were scattered with a predetermined resolution interval, and the LS sequence was employed as a template to generate the matching sequence. If the number of LS sequence

measurement points in the calculation is too large in the first stage, it will affect the matching efficiency. If the number is too low, it will reduce the probability of a successful matching. Thus, eight points were used in a single particle sequence as $MS_i = \{MS_{i1}, MS_{i2}, MS_{i3}, \dots, MS_{i8}\}$, where $MS_{ij} = (B_{ij}, L_{ij}, \partial \tilde{g}_{ij})$ and $\partial \tilde{g}_{ij}$ are the interpolated gravity anomalies at (B_{ij}, L_{ij}) . The PSO algorithm was utilized to obtain the starting point.

The search in the first stage is a necessary post-hoc process, as sufficient feature information cannot be obtained from only a single point, and the results are often divergent. MS_i is the rigid transformation of $\{LS_1, LS_2, LS_3, \dots, LS_8\}$ that neglects the noise

and Transitions between tracks. According to the fundamental concept of inertial navigation, the difference between the real and inertial tracks is negligible during the initial stage of error accumulation, as illustrated in Figure 1. LS_i , $i = 1, 2, 3, \dots, n$, and hence the influence of noise and changes between tracks was ignored. With an increase in the number of sample points, MS_i and LS_i changed considerably, as illustrated by point LS_{n+j} , $j = 1, 2, 3, \dots$ in Figure 1. Moreover, in order to match the coordinates of the gravity point in real time, the strategy must be adjusted in the second stage.

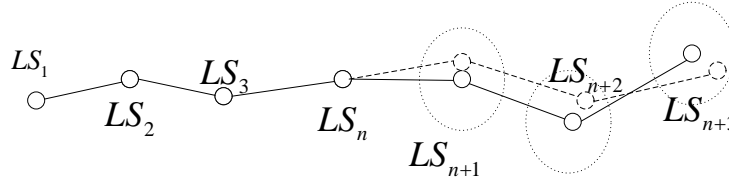


Figure 1. Schematic diagram of an inertial navigation track and a matching track

In the second stage, the starting points, B_0, L_0 , were swapped into the particle, assuming $MST_i = (B_0, L_0, \theta_i, \partial B_i, \partial L_i)$ $i = 1, 2, \dots, L$, where θ_i is the rotation angle, and φ is the scaling factor. The matching track and the real track share several similarities, and according to the affine transformation, the point I is expressed as:

$$\begin{bmatrix} B_i \\ L_i \end{bmatrix} = \varphi \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \Delta B \\ \Delta L \end{bmatrix} + \begin{bmatrix} B_0 \\ L_0 \end{bmatrix} \quad (10)$$

where ΔB and ΔL are translation parameters. The affine transformation was then applied to $LS = \{LS_1, LS_2, LS_3, \dots, LS_{i-1}\}$ and the optimal position of point i was estimated using the PSO algorithm.

4. SIMULATION PREPARATION

The steps utilized to implement the PSO algorithm are as follows:

- 1) Define a search area in which the starting point of the inertial navigation system is located in the center and the radius is set as the maximum allowable navigation error of 3σ .
- 2) Initialize the particle swarm by setting the parameters of the first and second stages, including the maximum number of

iterations, learning factor, position and speed, and fitness function.

- 3) Determine the fitness value of the initial particle in order to obtain the initial local and global optimal particles.

- 4) Update the speed and position of the particle and specify the fitness value of each particle, which is then compared to the existing optimum of the individual particle. If the MSD value is small, the local optimal particle is replaced with the current particle.

- 5) Compare the fitness value of each particle to the global optimum and store the particle with the lowest MSD value as the new global optimal particle.

- 6) Exit when the terminating condition is met (i.e., the error is acceptable or the maximum number of iterations is reached), otherwise return to Step 4, and increase the iteration number by one.

Two sea areas were selected in this study, and two tracks of underwater vehicles were obtained through simulation. Table 1 reports the coordinates of the starting point, the initial error, the heading angle, the speed, the linear error, and the random error of the simulated tracks. The reference map for gravity anomalies was the DTU12 model released by the Danish University of Science and Technology, the model has a 1' resolution and an accuracy of 3-8mGal.

Table 1. Simulated tracks and parameters

Parameters	Track 1	Track2
Coordinates of the starting point of the simulated real tracks	18.5 °N, 114 °E	26.5 °S, 161 °E
Initial error of the INS tracks	1n mile, 1n mile	-2n mile, 2n mile
Heading angle of the simulated real tracks	50 °	Change
Velocity of the simulated real tracks		10n mile/h
INS linear error		2n mile/h
Random error of the INS tracks		0.12n mile
Root mean square error of track point gravity anomaly		1mGal

Sampling number of INS tracks
Random error of the simulated real tracks

160
0.06n mile

5. EXPERIMENTAL VALIDATION AND ANALYSIS

The PSO algorithm was employed to carry out matching navigation for the two simulated tracks. The results are illustrated in Figures 2 and 3, with Figure 2 depicting a straight track, and Figure 3 depicting a curved track. As can be observed, the matching and real tracks were quite consistent. A gravity anomaly was interpolated between the matching track points. The longitude, latitude, and interpolated gravity anomaly of the matched track points were compared to the real track. Tables 1 and 2 report the maximum, minimum, average, and standard deviation of the differences in longitude, latitude, and gravity anomalies. It can be noticed that the difference was within 1n

mile, the average difference in gravity anomaly was within 0.4mGal, and the standard deviation was within 1.5mGal.

In terms of computational time, the matching time was identical for the two tracks because they had the same number of track points, iterations, and particles. The matching speed in the first stage is quick. With 200 iterations and 100 particles, the calculation took 196 seconds. As the number of track points increased in the second stage of calculation, the computational time for each track point continuously increased proportionally. When there are many matching track points, the track can be calculated in segments to improve the timeliness.

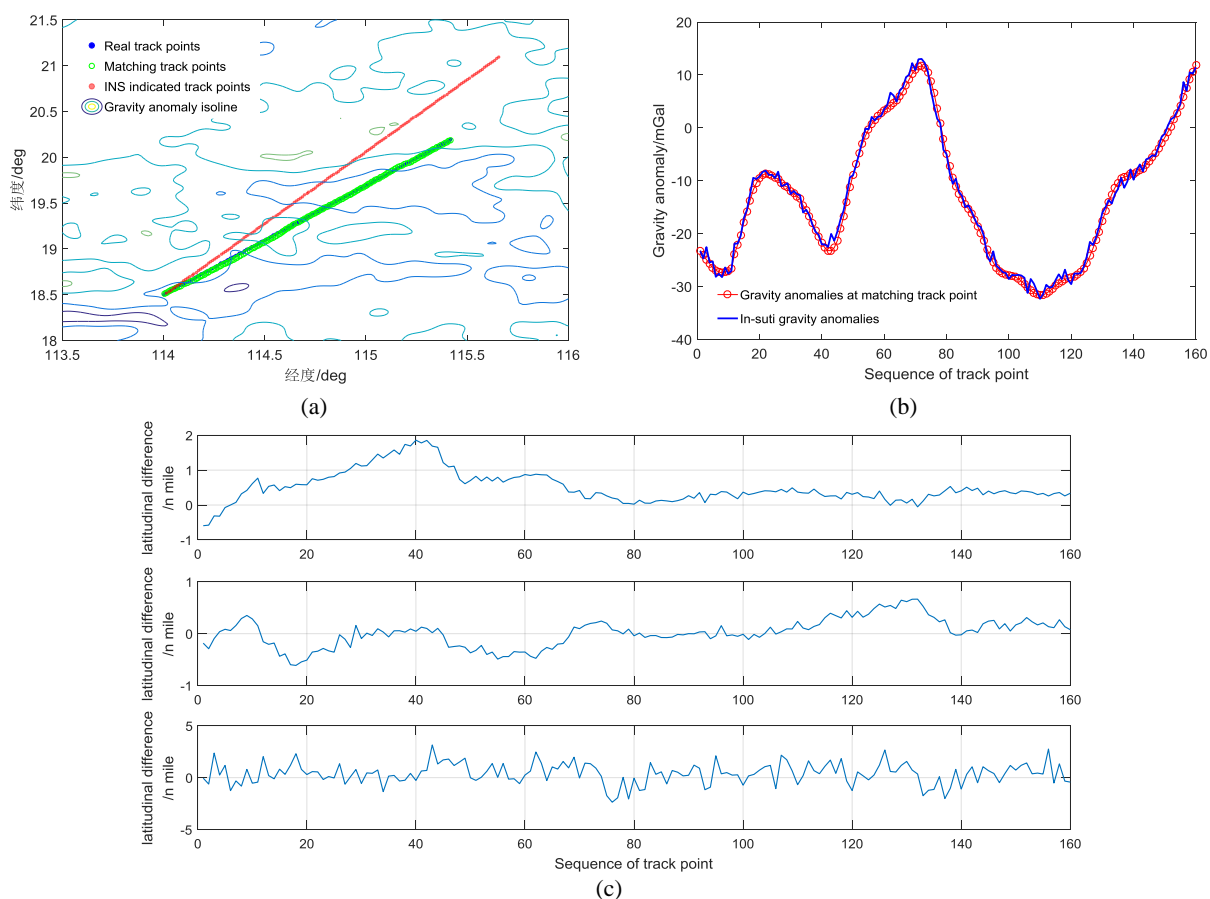


Figure 2. Matching results of track I

Table 2. The accuracy of matching navigation results

	Minimum	Maximum	Average	Standard deviation
Latitudinal difference	-0.59n mile	1.86n mile	0.50n mile	0.45n mile
longitudinal difference	-0.69n mile	0.67n mile	0.02n mile	0.27n mile
Difference in gravity anomaly	-2.37mGal	3.15mGa	0.40mGal	1.01mGal

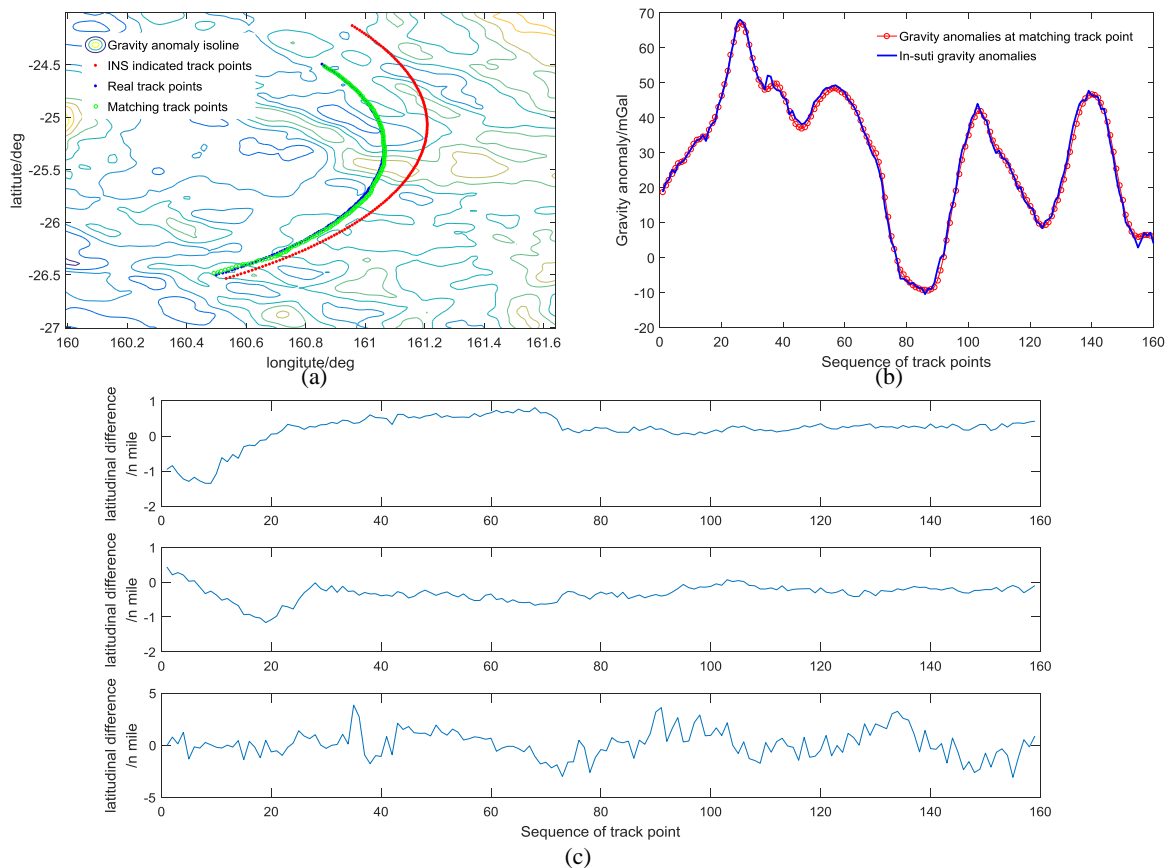


Figure 3. Matching results of track II

Table 3. The accuracy of matching navigation results

	Minimum	Maximum	Average	Standard deviation
Latitudinal difference	-1.35n mile	0.81n mile	0.19n mile	0.43n mile
longitudinal difference	-1.39n mile	0.44 n mile	-0.32n mile	0.24n mile
Difference in gravity anomaly	-3.08mGal	3.84mGal	0.15mGal	1.33mGal

6. CONCLUSIONS

In this study, the PSO algorithm was integrated with affine transformation to simulate underwater gravity-matching navigation. Following a review of the principles and implementation of the PSO algorithm, a two-stage solution was proposed for underwater gravity matching navigation. The first stage was used to determine the starting point, and the second stage was to match the real-world track points in real time, which not only ensured the stability of the results, but also took into account the real-time performance and effectiveness. The results indicated that the matching track and the real track were in good agreement, and that the differences in the position and gravity anomaly between the two tracks remained within a reasonable range. In terms of computational time, the calculation efficiency decreased as the number of track points increased. The authors will attempt to segment the track and perform matching navigation separately in their future research.

REFERENCES

Chen xianghong.,Zhou yuehua.,2017:Underwater terrain-aided navigation method based on improved particle swarm

optimization[J].Journal of Chinese Inertial Technology. 25(6):770-77.

Golubovic R M.,Olcan D I.,Kolundzija B M.,2007:Particle Swarm Optimization Algorithm and Its Modifications Applied to EM Problems[C].International Conference on Telecommunications in Modern Satellite, Cable & Broadcasting Services. IEEE,

Grandi R.,Falconi R.,Melchiorri C.,2012:A Navigation Strategy for Multi-Robot Systems Based on Particle Swarm Optimization Techniques[J].Ifac Proceedings Volumes, 45(22):331-336.

Han Y.,Wang B.,Deng Z.,2016:An Improved TERCOM-Based Algorithm for Gravity-Aided Navigation[J].IEEE Sensors Journal, 16(8): 2537-2544.

HUANG Peng.,CHENG Yi.,2011:Application of Improved Back Propagation Neural Network Algorithm in Track Matching[J].COMPUTER ENGINEERING, 37(11): 218-219.

Liu Chengxiang.,Liu fanming.,Liu zhu.,2003:Study on Technique for Seabottom Terrain Mapping System by Means of Rapid ICCP Algorithm[J].SHIP ENGINEERING, (03): 54-56.

LIU Wei.,WU Zhitian.,WU Meiping.,HU Xiaoping.,2011: Geomagnetic Matching Technology Based on Improved ICCP Matching Algorithm[J].Navigation and Control , 10(2):1-3,21

Wang B.,Yu L.,Deng Z.,2016:A Particle Filter-Based Matching Algorithm With Gravity Sample Vector for Underwater Gravity Aided Navigation[J].ieee/asme transactions on mechatronics, 21(3):1399-1408.

Wang B.,Zhu J.,Deng Z.,2019:A Characteristic Parameter Matching Algorithm for Gravity-Aided Navigation of Underwater Vehicles[J]. Industrial Electronics, IEEE Transactions on, 66(2):1203-1212.

WANG shengping.,ZHANG hongmei.,ZHAO Jianhu.,TANG Ping.,2011:Marine Geomagnetic Navigation Technology Based on Integration of TERCOM and ICCP[J].Geomatics and Information Science of Wuhan University.36(10): 1209-1212.

WANG Xianglei.,SU Mudan.,LIU Peigen.,DING Shuo., 2013:Application of improved ICCP algorithm in gravity matching aided navigation[J].Science of Surveying and Mapping, 38(01):36-39+50.

Wei E.,Dong C.,Yang Y.,2017:A Robust Solution of Integrated SITAN with TERCOM Algorithm: Weight Reducing Iteration Technique for Underwater Vehicles' Gravity Aided Inertial Navigation System[J].Navigation,64(1): 111-122.

Wei E.,Dong C.,Yang Y.,2017:A Robust Solution of Integrated SITAN with TERCOM Algorithm: Weight Reducing Iteration Technique for Underwater Vehicles' Gravity Aided Inertial Navigation System[J].Navigation,64(1): 111-122.

WEI Erhu.,DONG Cuijun.,LIU Jiandong.,TANG Shenquan.,2017:An Improved TERCOM Algorithm for Gravity-Aided Inertial Navigation System[J].Journal of Geomatics,42(6):29-31,100.

Yang yuanxi.,Xu tianhe.,Xue shuqiang.,2017:Progresses and Prospects in Developing Marine Geodetic Datum and Marine Navigation of China[J]. Acta Geodaetica et Cartographica Sinica,(01):5-12.

Yang Zhenli.,Zhu zhuangsheng.,2014:A Triangle Matching Algorithm for Gravity-aided Navigation for Underwater Vehicles.[J]. Journal of Navigation.

YUAN Gannan.,TAN Jialin.,YIN Weiwei.,2010:Seabed Terrain Aided Navigation Algorithm Based on Hybrid Particle Swarm Optimization[J]. COMPUTER SIMULATION,(08):128-132.