frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 08 May 2019
doi: 10.3389/fnbot.2019.00019

Solving Gravity Anomaly Matching Problem Under Large Initial Errors in Gravity Aided Navigation by Using an Affine Transformation Based Artificial Bee Colony Algorithm

Tian Dai, Lingjuan Miao, Haijun Shao and Yongsheng Shi*

School of Automation, Beijing Institute of Technology, Beijing, China

Gravity aided inertial navigation system (GAINS), which uses earth gravitational anomaly field for navigation, holds strong potential as an underwater navigation system. The gravity matching algorithm is one of the key factors in GAINS. Existing matching algorithms cannot guarantee the matching accuracy in the matching algorithms based gravity aided navigation when the initial errors are large. Evolutionary algorithms, which are mostly have the ability of global optimality and fast convergence, can be used to solve the gravity matching problem under large initial errors. However, simply applying evolutionary algorithms to GAINS may lead to false matching. Therefore, in order to deal with the underwater gravity matching problem, it is necessary to improve the traditional evolutionary algorithms. In this paper, an affine transformation based artificial bee colony (ABC) algorithm, which can greatly improve the positioning precision under large initial errors condition, is developed. The proposed algorithm introduces affine transformation to both initialization process and evolutionary process of ABC algorithm. The single-point matching strategy is replaced by the strategy of matching a sequence of several consecutive position vectors. In addition, several constraints are introduced to the process of evolution by using the output characteristics of the inertial navigation system (INS). Simulations based on the actual gravity anomaly base map have been performed for the validation of the proposed algorithm.

Keywords: gravity aided navigation, bio-inspired navigation, navigation systems, optimization, underwater vehicle, evolutionary algorithm

OPEN ACCESS

Edited by:
Xiaofeng Xiong,
University of Southern Denmark,
Denmark

Reviewed by:
Ning Sun,
Nankai University, China
Chin-Shiuh Shieh,
National Kaohsiung First University of
Science and Technology, Taiwan

***Correspondence:**
Tian Dai
daitian@bit.edu.cn

Received: *16 November 2018*
Accepted: *17 April 2019*
Published: *08 May 2019*

Citation:
Dai T, Miao L, Shao H and Shi Y (2019) Solving Gravity Anomaly Matching Problem Under Large Initial Errors in Gravity Aided Navigation by Using an Affine Transformation Based Artificial Bee Colony Algorithm. Front. Neurorobot. 13:19. doi: 10.3389/fnbot.2019.00019

INTRODUCTION

It is well known that inertial navigation systems (INSs) typically used on underwater vehicles tend to develop accumulated errors. Above water, the INS data can be corrected by the use of global navigation satellite system (GNSS) (Bishop, 2002). However, due to the rapid attenuation of higher frequency signals and the unstructured nature of the undersea environment, GNSS signals can only propagate within short distance under water (Paull et al., 2013). As a passive navigation system, the gravity aided inertial navigation system (GAINS) holds strong potential as an auxiliary navigation system (Canciani and Raquet, 2017). GAINS obtains a gravity anomaly measurements by using gravimeters installed on a vehicle. These measurements are matched with a priori digital map of the gravity anomaly, to estimate the vehicle position. GAINSs are completely passive and difficult to interfere with. These advantages are totally meet the requirement of underwater vehicles (Rice et al., 2004).

A GAINS consists of a priori gravity anomaly database (gravity anomaly map), a measurement unit (gravimeter), and a navigation algorithm. There are two types of GAINSs: filtering algorithms based GAINSs (Li et al., 2013; Claus and Bachmayer, 2015; Copp and Subbarao, 2015; Allotta et al., 2016) and matching algorithms based GAINSs (Zhao et al., 2014; Wu et al., 2015; Zhu et al., 2015; Han et al., 2016; Song et al., 2016). Filtering algorithms, which take use of simplified dynamic mathematical model of the vehicle, have a good performance in real-time. In filtering algorithms based GAINSs, commonly used filters are the extended Kalman filter (EKF), the particle filter (PF) and the Rao-Blackwellized particle filter (RBPF). EKF is more effective for low nonlinear estimation problems. PF can effectively handle highly non-linear or non-Gaussian estimation problems. RBPF is a hybrid filter combining EKF and PF (Simanek et al., 2015; Kim T. et al., 2018; Kim Y. et al., 2018). However, the application of filtering algorithms is limited since the precise model of gravity anomaly is difficult to establish (Han et al., 2016). The matching algorithm is a key factor in matching algorithms based GAINSs (Hegrenaes and Hallingstad, 2011; Wu et al., 2017). Terrain contour matching (TERCOM) (Affleck and Jircitano, 1990) algorithm and iterative closest contour point (ICCP) (Kamgarparsi and Kamgarparsi, 1999) algorithm are two conventional matching algorithms in GAINSs. TERCOM algorithm is realized via group correlation analysis. Most of the improvements to the TERCOM algorithm (Zhao et al., 2014; Han et al., 2016) are proposed to solve bad real-time performance and heavy computational complexity problem. However, TERCOM algorithm is sensitive to angular error of the INS-indicated segment, which is difficult to improve. ICCP algorithm, which uses rigid transformation to matching the multilateral arc, has a high matching accuracy. Over the past decade, a number of improved ICCP algorithm have been suggested. Among them, a part of researches were developed for improving the real-time performance (Tong et al., 2011; Wang et al., 2013). Other studies adopted affine transformation to deal with the scale error (Xu et al., 2014; Song et al., 2016). However, the mismatching of ICCP algorithm is easily included when the initial position errors of INS are large (Han et al., 2016). Both TERCOM-based algorithms and ICCP-based algorithms cannot guarantee the matching accuracy when initial errors are large. TERCOM-based algorithms are sensitive to angular errors of the INS-indicated segments and ICCP-based algorithms are sensitive to position errors. Therefore, a matching algorithm that maintain high matching precision under large initial errors is needed. Evolutionary algorithms (Quan and Fang, 2010; Gao et al., 2014; Hidalgo-Paniagua et al., 2016; Teymourian et al., 2016; Li et al., 2017) are a good choice because most of them have the ability of both global optimality and fast convergence.

The evolutionary algorithms have emerged as a powerful tool for finding optimum solutions of complex optimization problems. In the past few decades, a number of evolutionary algorithms have been used extensively to obtain optimal designs and overcome the computational drawbacks of traditional mathematical optimization methods (Abrao, 2013; Yildiz, 2013). Jaesung and Kim improved the genetic algorithm so solve robot path planning problem (Lee and Kim, 2016). An improved intelligent water drops algorithm and an advanced cuckoo search algorithm are proposed to solve the capacitated vehicle routing problem (Teymourian et al., 2016). Improved bee colony algorithm is applied to the color image segmentation, assigning the optimal coordinates of seeds and determining similarity differences (Sag and Çunkaş, 2015).

Gravity matching problem can be regarded as an optimization problem (Wu et al., 2017). Therefore, evolutionary algorithms can be directly applied to the gravity matching problem. Gao et al. proposed an improved artificial bee colony (ABC) algorithm and obtained a good performance on gravity matching under small initial errors (Gao et al., 2014). However, there exists a lot of similarity feature points in matching area when initial errors are large. Although the improved ABC algorithm can be implemented on navigation systems under large initial errors, the mismatching of position is easily included. In this paper, we consider a challenging gravity matching problem of achieving high precision under large initial errors. To solve this problem, we propose an affine transformation based ABC algorithm. Firstly, In order to avoid mismatching caused by similarity feature points, we apply sequence matching strategy to the proposed algorithm. Secondly, Scaling transformation, rotation transformation, and translation transformation are introduced to both initialization process and evolutionary process to increase the convergence speed. In addition, we constrain the affine transformation by utilizing the output characteristics of INS.

The rest of the paper is organized as follows. The second part introduces the ABC algorithm. INS error propagation model and the constraints of affine transformation are provided in the third part. The fourth part of this paper presents the procedure of affine transformation based artificial bee colony algorithm. A comprehensive discussion on the experimental settings and simulation results are provided in the fifth part. Conclusions are given in the last part.

ARTIFICIAL BEE COLONY ALGORITHM

The ABC algorithm (Karaboga and Basturk, 2007) is one of the most recently introduced swarm-based search methods (Akay and Karaboga, 2012; Akbari et al., 2012; Sag and Çunkaş, 2015; Li et al., 2016). It contains three groups of bees: employed bees, onlookers, and scouts. In ABC algorithm, the position of a food source represents a possible solution of the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. The more a solution has high fitness, the more possibility of being selected of this solution by onlooker bees. The number of the employed bees or the onlooker bees is equal to the number of solutions in the population. The initial swarm composed of employed bees are generated by (1):

$$\boldsymbol{x}_i^j = \boldsymbol{x}_{\min}^j + rand(0,1)(\boldsymbol{x}_{\max}^j - \boldsymbol{x}_{\min}^j) \tag{1}$$

Where $\boldsymbol{x}_i^j$ indicates the jth parameter of ith solution in the population, $i = \{1, 2, \cdots, SN\}$ and SN is the size of population, $j = \{1, 2, \cdots, D\}$ and D is the number of parameters in a solution. $\boldsymbol{x}_{\min}^j$ and $\boldsymbol{x}_{\max}^j$ are, respectively, the lower bound and the upper bound of jth parameter. $rand(0, 1)$ generates a real number between 0 and 1.

After initialization, fitness values of solutions are calculated by (2):

$$fit_i = \begin{cases} 1/(1+f_i) & f_i \geq 0 \\ 1 + |f_i| & f_i < 0 \end{cases} \tag{2}$$

Where f_i is the objective function value of ith solution in the population. fit_i is the fitness value of ith solution in the population. The employed bee evaluates the quality of food sources and determines a new one according to the fitness value. If the nectar amount of new source is better than the old source, employed bee will update its memory with new source. The employed bees update their sources according to the following equation:

$$\boldsymbol{v}_i^j = \boldsymbol{x}_i^j + rand(-1, 1)(\boldsymbol{x}_i^j - \boldsymbol{x}_{neighbor}^j) \tag{3}$$

Where $\boldsymbol{x}_{neighbor}^j$ is the neighbor solution selected randomly, neighbor $\in \{1, 2, \cdots, i-1, i+1, \cdots, SN\}$. $\boldsymbol{v}_i^j$ represents new solution.

After new food sources have been explored, onlookers select an employed bee for guidance. For this purpose, roulette wheel is used to calculate the probabilities:

$$p_i = \frac{fit_i}{\sum\limits_{q=1}^{SN} fit_q} \tag{4}$$

After the onlooker bees select a food source as a guide, the candidate food source is calculated by (3). Afterwards, the greedy selection process is applied for the onlooker bees. The best solution achieved so far is memorized. In addition, a food source will be abandoned when limit, a control parameter, is exceeded for the source. So it is replaced with randomly produced solution by (1).

For the gravity matching problem, denote $\boldsymbol{x}$ as the position of vehicle and f as the gravity objective function. Then ABC algorithm can be directly used to find the optimal position. However, simply applying ABC algorithm to the gravity matching problem may lead to false matching (Gao et al., 2014). There exists a lot of similarity gravity feature points in gravity database, which means single-point matching process may easily convergence at the wrong position.

SEQUENCE MATCHING STRATEGY

To solve the mismatching problem, the single-point matching strategy in basic ABC is replaced by strategy of matching a sequence of several consecutive position vectors. Considering the short-term high-precision characteristics of the INS (Wu et al., 2015), affine transformation could be used to describe the relationship between the sequence to be matched and the INS-indicated sequence. Specifically, these transformations include scaling transformation, rotation transformation, and translation transformation. The range of affine transformations can be estimated by an INS error propagation model.

INS Error Propagation Model

Affine transformation is constrained by the output characteristics of the INS. Therefore, an INS error propagation model (Yan et al., 2008) is employed in this paper. In Yan et al. (2008), i-reference frame is the fixed inertial frame; e-reference frame is the Terrestrial reference frame (TRF); n-reference frame has its origin on the surface and its axes pointing East, North, and Up (ENU reference frame); b-reference frame is centered in the center of gravity of the vehicle, with the y-axis pointing in the direction of the forward motion of the vehicle, the z-axis pointing up and the x-axis completing a right-handed reference frame. However, GAINSs are commonly used in underwater environments. In which case n-reference frame adopts North, East and Down (NED) reference frame; b-reference frame is centered in the center of gravity of the vehicle, with the x-axis pointing in the direction of the forward motion of the vehicle, the z-axis pointing down and the y-axis completing a right-handed reference frame (Allotta et al., 2016). Hence the INS error propagation model in Yan et al. (2008) needs to be modified.

In actual case, there exist rotation errors between n-reference frame (ideal mathematics platform) and n'-reference frame (actual mathematics platform). n'-reference frame can be gained after rotating 3 times of n-reference frame. Let α_z, α_y, α_x be the three rotational angles and $\boldsymbol{\alpha} = \begin{bmatrix} \alpha_x & \alpha_y & \alpha_z \end{bmatrix}$, the rotation matrixes can be expressed as:

$$\begin{aligned} \boldsymbol{C}_{\alpha_z} &= \begin{bmatrix} \cos\alpha_z & -\sin\alpha_z & 0 \\ \sin\alpha_z & \cos\alpha_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \boldsymbol{C}_{\alpha_y} &= \begin{bmatrix} \cos\alpha_y & 0 & -\sin\alpha_y \\ 0 & 1 & 0 \\ \sin\alpha_y & 0 & \cos\alpha_y \end{bmatrix} \\ \boldsymbol{C}_{\alpha_x} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha_x & \sin\alpha_x \\ 0 & -\sin\alpha_x & \cos\alpha_x \end{bmatrix} \end{aligned} \tag{5}$$

Then, the coordinate transformation matrix from n-reference frame to n'-reference frame is derived:

$$\boldsymbol{C}_{\mathrm{n}}^{\mathrm{n}'} = \boldsymbol{C}_{\alpha_x} \boldsymbol{C}_{\alpha_y} \boldsymbol{C}_{\alpha_z} \tag{6}$$

Denoting with $\boldsymbol{\omega}_{\mathrm{nn}'}^{\mathrm{n}'}$ the relative angular velocity of n'-reference frame in n-reference frame, the following equation has been derived:

$$\boldsymbol{\omega}_{\mathrm{nn}'}^{\mathrm{n}'} = \boldsymbol{C}_{\alpha_x} \boldsymbol{C}_{\alpha_y} \begin{bmatrix} 0 \\ 0 \\ \dot{\alpha}_z \end{bmatrix} + \boldsymbol{C}_{\alpha_x} \begin{bmatrix} 0 \\ \dot{\alpha}_y \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\alpha}_x \\ 0 \\ 0 \end{bmatrix} = \boldsymbol{C}_{\omega} \begin{bmatrix} \dot{\alpha}_x \\ \dot{\alpha}_y \\ \dot{\alpha}_z \end{bmatrix} \tag{7}$$

Hence, the differential equation of Euler platform error angles can be expressed as follows:

$$\dot{\boldsymbol{\alpha}}=\boldsymbol{C}_{\omega}^{-1}\boldsymbol{\omega}_{\mathrm{nn'}}^{\mathrm{n'}} \tag{8}$$

Here $\mathbf{C}_{\omega}$ and $\mathbf{C}_{\omega}^{-1}$ are calculated by the following expressions:

$$\begin{aligned}\boldsymbol{C}_{\omega}&=\begin{bmatrix}1 & 0 & -\sin\alpha_y\\ 0 & \cos\alpha_x & \sin\alpha_x\cos\alpha_y\\ 0 & -\sin\alpha_x & \cos\alpha_x\cos\alpha_y\end{bmatrix}\\ \boldsymbol{C}_{\omega}^{-1}&=\frac{1}{\cos\alpha_y}\begin{bmatrix}\cos\alpha_y & \sin\alpha_x\sin\alpha_y & \cos\alpha_x\sin\alpha_y\\ 0 & \cos\alpha_x\cos\alpha_y & -\sin\alpha_x\cos\alpha_y\\ 0 & \sin\alpha_x & \cos\alpha_x\end{bmatrix}\end{aligned} \tag{9}$$

Denoting with $\boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{n}}$ the rotational angular velocity of the earth. L and h are, respectively, latitude, and depth. R_{M} is the local radius of curvature in meridian and R_{N} is the local radius of curvature in prime vertical. $\hat{L}=L+\delta L$, $\hat{h}=h+\delta h$. In addition, δL and δh are slight errors. The following equations have been derived:

$$\boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{n}}=\left[\omega_{\mathrm{ie}}\cos L\;\;0\;\;-\omega_{\mathrm{ie}}\sin L\right]^{\mathrm{T}} \tag{10}$$

$$\boldsymbol{\omega}_{\mathrm{en}}^{\mathrm{n}}=\left[\frac{v_{\mathrm{E}}^{\mathrm{n}}}{R_{\mathrm{N}}-h}\;\;-\frac{v_{\mathrm{N}}^{\mathrm{n}}}{R_{\mathrm{M}}-h}\;\;-\frac{v_{\mathrm{E}}^{\mathrm{n}}}{R_{\mathrm{N}}-h}\tan L\right]^{\mathrm{T}} \tag{11}$$

$$\delta\boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{n}}=\begin{bmatrix}-\omega_{\mathrm{ie}}\sin\hat{L}\delta L\\ 0\\ -\omega_{\mathrm{ie}}\cos\hat{L}\delta L\end{bmatrix} \tag{12}$$

$$\delta\boldsymbol{\omega}_{\mathrm{en}}^{\mathrm{n}}=\begin{bmatrix}\delta v_{\mathrm{E}}^{\mathrm{n}}/\left(\hat{R}_{\mathrm{N}}-\hat{h}\right)\\ -\delta v_{\mathrm{N}}^{\mathrm{n}}/\left(\hat{R}_{\mathrm{M}}-\hat{h}\right)\\ -\left(\tan\hat{L}\delta v_{\mathrm{E}}^{\mathrm{n}}+\hat{v}_{\mathrm{E}}^{\mathrm{n}}\sec^2\hat{L}\delta L\right)/\left(\hat{R}_{\mathrm{N}}-\hat{h}\right)\end{bmatrix} \tag{13}$$

$\boldsymbol{\omega}_{\mathrm{in}}^{\mathrm{n}}$ and $\delta\boldsymbol{\omega}_{\mathrm{in}}^{\mathrm{n}}$ can be expressed as follows:

$$\omega_{\mathrm{in}}^{\mathrm{n}}=\boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{n}}+\boldsymbol{\omega}_{\mathrm{en}}^{\mathrm{n}} \tag{14}$$

$$\delta\boldsymbol{\omega}_{\mathrm{in}}^{\mathrm{n}}=\delta\boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{n}}+\delta\boldsymbol{\omega}_{\mathrm{en}}^{\mathrm{n}} \tag{15}$$

On the basis of these formulas, the INS attitude error and velocity error have been derived (Yan et al., 2008):

$$\dot{\boldsymbol{\alpha}}=\boldsymbol{C}_{\omega}^{-1}\left[\left(\mathbf{I}\text{-}\boldsymbol{C}_{\mathrm{n}}^{\mathrm{n'}}\right)\hat{\boldsymbol{\omega}}_{\mathrm{in}}^{\mathrm{n}}+\boldsymbol{C}_{\mathrm{n}}^{\mathrm{n'}}\delta\boldsymbol{\omega}_{\mathrm{in}}^{\mathrm{n}}-\boldsymbol{C}_{\mathrm{b}}^{\mathrm{n'}}\delta\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}}\right] \tag{16}$$

$$\begin{aligned}\delta\dot{\boldsymbol{v}}^{\mathrm{n}}=&\left[\boldsymbol{I}\text{-}\left(\boldsymbol{C}_{\mathrm{n}}^{\mathrm{n'}}\right)^{\mathrm{T}}\right]\boldsymbol{C}_{\mathrm{b}}^{\mathrm{n'}}\hat{\boldsymbol{f}}_{\mathrm{sf}}^{\mathrm{b}}+\left(\boldsymbol{C}_{\mathrm{n}}^{\mathrm{n'}}\right)^{\mathrm{T}}\boldsymbol{C}_{\mathrm{b}}^{\mathrm{n'}}\delta\boldsymbol{f}_{\mathrm{sf}}^{\mathrm{b}}-\left(2\delta\boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{n}}+\delta\boldsymbol{\omega}_{\mathrm{en}}^{\mathrm{n}}\right)\\ &\times\left(\hat{\boldsymbol{v}}^{\mathrm{n}}-\delta\boldsymbol{v}^{\mathrm{n}}\right)-\left(2\hat{\boldsymbol{\omega}}_{\mathrm{ie}}^{\mathrm{n}}+\hat{\boldsymbol{\omega}}_{\mathrm{en}}^{\mathrm{n}}\right)\times\delta\boldsymbol{v}^{\mathrm{n}}+\delta\boldsymbol{g}^{\mathrm{n}}\end{aligned} \tag{17}$$

Where $\delta\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}}$ is the measurement error of gyroscope, $\delta\boldsymbol{\omega}_{\mathrm{in}}^{\mathrm{n}}$ is the calculation error of $\boldsymbol{\omega}_{\mathrm{in}}^{\mathrm{n}}$. $\delta\boldsymbol{f}_{\mathrm{sf}}^{\mathrm{b}}$ is the measurement error of accelerometer. In addition, $\delta\boldsymbol{\omega}_{\mathrm{ib}}^{\mathrm{b}}$ are mainly consist of a constant bias $\boldsymbol{\varepsilon}^{\mathrm{b}}$ and zero mean Gaussian white noise $\mathbf{w}_{\mathrm{g}}^{\mathrm{b}}$. $\delta\boldsymbol{f}_{\mathrm{sf}}^{\mathrm{b}}$ are mainly consist of a constant bias ∇^{b} and zero mean Gaussian white noise $\mathbf{w}_{\mathrm{a}}^{\mathrm{b}}$. Besides, $\delta\boldsymbol{g}^{\mathrm{n}}$could be ignored. Hence, the INS error propagation model used in underwater environment is defined by the following equations:

$$\begin{cases}\delta\dot{L}=\dfrac{\delta v_{\mathrm{N}}^{\mathrm{n}}}{R_{\mathrm{M}}-h}+\delta h\dfrac{v_{\mathrm{N}}^{\mathrm{n}}}{(R_{\mathrm{M}}-h)^2}\\ \delta\dot{\lambda}=\dfrac{\delta v_{\mathrm{E}}^{\mathrm{n}}}{R_{\mathrm{N}}-h}\sec L+\delta L\dfrac{v_{\mathrm{E}}^{\mathrm{n}}}{R_{\mathrm{N}}-h}\tan L\sec L+\delta h\dfrac{v_{\mathrm{E}}^{\mathrm{n}}\sec L}{(R_{\mathrm{N}}-h)^2}\\ \delta\dot{h}=\delta V_{\mathrm{D}}\\ \dot{\boldsymbol{\alpha}}=\boldsymbol{C}_{\omega}^{-1}\left[\left(\boldsymbol{I}-\boldsymbol{C}_{\mathrm{n}}^{\mathrm{n'}}\right)\hat{\boldsymbol{\omega}}_{\mathrm{in}}^{\mathrm{n}}+\boldsymbol{C}_{\mathrm{n}}^{\mathrm{n'}}\delta\boldsymbol{\omega}_{\mathrm{in}}^{\mathrm{n}}-\boldsymbol{C}_{\mathrm{b}}^{\mathrm{n'}}\boldsymbol{\varepsilon}^{\mathrm{b}}\right]-\boldsymbol{C}_{\omega}^{-1}\boldsymbol{C}_{\mathrm{b}}^{\mathrm{n'}}\boldsymbol{w}_{\mathrm{g}}^{\mathrm{b}}\\ \delta\dot{\boldsymbol{v}}^{\mathrm{n}}=\left[\boldsymbol{I}-\left(\boldsymbol{C}_{\mathrm{n}}^{\mathrm{n'}}\right)^{\mathrm{T}}\right]\boldsymbol{C}_{\mathrm{b}}^{\mathrm{n'}}\hat{\boldsymbol{f}}_{\mathrm{sf}}^{\mathrm{b}}+\left(\boldsymbol{C}_{\mathrm{n}}^{\mathrm{n'}}\right)^{\mathrm{T}}\boldsymbol{C}_{\mathrm{b}}^{\mathrm{n'}}\nabla^{\mathrm{b}}-\left(2\delta\boldsymbol{\omega}_{\mathrm{ie}}^{\mathrm{n}}+\delta\boldsymbol{\omega}_{\mathrm{en}}^{\mathrm{n}}\right)\\ \times\left(\hat{\boldsymbol{v}}^{\mathrm{n}}-\delta\boldsymbol{v}^{\mathrm{n}}\right)-\left(2\hat{\boldsymbol{\omega}}_{\mathrm{ie}}^{\mathrm{n}}+\hat{\boldsymbol{\omega}}_{\mathrm{en}}^{\mathrm{n}}\right)\times\delta\boldsymbol{v}^{\mathrm{n}}+\left(\boldsymbol{C}_{\mathrm{n}}^{\mathrm{n'}}\right)^{\mathrm{T}}\boldsymbol{C}_{\mathrm{b}}^{\mathrm{n'}}\boldsymbol{w}_{\mathrm{a}}^{\mathrm{b}}\\ \dot{\boldsymbol{\varepsilon}}^{\mathrm{b}}=0\\ \dot{\nabla}^{\mathrm{b}}=0\end{cases} \tag{18}$$

Range of Scaling Transformation

Through the INS error propagation model (18), the range of rotation transformation and translation transformation can be estimated. However, the range of scaling transformation cannot be directly calculated. To better describe the scaling transformation, the INS-indicated distance and the actual distance in a short period are expressed by the following equations:

$$d_{k-1,k}^{\mathrm{INS}}=\frac{\left\|\boldsymbol{v}_{k-1}^{\mathrm{INS}}+\boldsymbol{v}_{k}^{\mathrm{INS}}\right\|}{2}\cdot\Delta t \tag{19}$$

$$d_{k-1,k}^{\mathrm{real}}\approx\frac{\left\|\boldsymbol{v}_{k-1}^{\mathrm{real}}+\boldsymbol{v}_{k}^{\mathrm{real}}\right\|}{2}\cdot\Delta t \tag{20}$$

$d_{k\text{-}1,k}^{\mathrm{INS}}$ and $d_{k-1,k}^{\mathrm{real}}$ are, respectively, the INS-indicated distance and the actual distance between k-1th sampling point and kth sampling point. $\boldsymbol{v}_{k}^{\mathrm{INS}}$ and $\boldsymbol{v}_{k}^{\mathrm{real}}$ are the INS-indicated linear velocity and the actual linear velocity of kth sampling point. Δt is the sampling period of the discrete time system.

Then, the relationship between $d_{k\text{-}1,k}^{\mathrm{INS}}$and $d_{k-1,k}^{\mathrm{real}}$ is derived:

$$\frac{d_{k-1,k}^{\mathrm{real}}}{d_{k-1,k}^{\mathrm{INS}}}=\frac{\left\|\boldsymbol{v}_{k-1}^{\mathrm{real}}+\boldsymbol{v}_{k}^{\mathrm{real}}\right\|}{\left\|\boldsymbol{v}_{k-1}^{\mathrm{INS}}+\boldsymbol{v}_{k}^{\mathrm{INS}}\right\|} \tag{21}$$

Where $\boldsymbol{v}_{k}^{\mathrm{INS}}$is described by an equation in the form:

$$\boldsymbol{v}_{k}^{\mathrm{INS}}=\boldsymbol{v}_{k}^{\mathrm{real}}+\delta\boldsymbol{v}_{k} \tag{22}$$

Here $\delta\boldsymbol{v}_k$ is the INS accumulated velocity error of kth sampling point. Accordingly, (21) is developed into the following form:

$$\frac{d_{k-1,k}^{\mathrm{real}}}{d_{k-1,k}^{\mathrm{INS}}}=\frac{\left\|\boldsymbol{v}_{k-1}^{\mathrm{INS}}+\boldsymbol{v}_{k}^{\mathrm{INS}}-\delta\boldsymbol{v}_{k\text{-}1}-\delta\boldsymbol{v}_{k}\right\|}{\left\|\boldsymbol{v}_{k-1}^{\mathrm{INS}}+\boldsymbol{v}_{k}^{\mathrm{INS}}\right\|} \tag{23}$$

Denoting with N the number of sampling points on matching sequence, the constraint of scale transform has been derived:

$$S=\frac{\sum_{k=2}^{N} d_{k-1,k}^{\text{real}}}{\sum_{k=2}^{N} d_{k-1,k}^{\text{INS}}}=\frac{\sum_{k=2}^{N}\left\|\boldsymbol{v}_{k-1}^{\text{INS}}+\boldsymbol{v}_{k}^{\text{INS}}-\delta\boldsymbol{v}_{k\text{-}1}-\delta\boldsymbol{v}_{k}\right\|}{\sum_{k=2}^{N}\left\|\boldsymbol{v}_{k-1}^{\text{INS}}+\boldsymbol{v}_{k}^{\text{INS}}\right\|} \tag{24}$$

AFFINE TRANSFORMATION BASED ABC ALGORITHM

In order to achieve high positioning accuracy under large initial errors, this paper proposes an affine transformation based ABC algorithm. The proposed algorithm is specifically presented to deal with the gravity matching problem for underwater navigation system. The affine transformation based ABC algorithm introduces affine transformation to both initialization process and evolutionary process of ABC algorithm. In addition, the affine transformation satisfies the constraint conditions provided by the output characteristics of the INS.

Objective Function and Initialization

The proposed algorithm use mean absolute difference (MAD) function as the objective function:

$$f(\boldsymbol{x})=\frac{1}{N}\sum_{k=1}^{N}\left|g_k^{\text{obs}}-g_k^{\boldsymbol{x}}\right| \tag{25}$$

Where $\boldsymbol{x}$ indicates a sequence of several consecutive position vectors. g_k^{obs} is the measured gravity anomaly value of kth sampling point and $g_k^{\boldsymbol{x}}$ is the corresponding gravity anomaly value on gravity anomaly base map of kth sampling point. $f(\boldsymbol{x})$ is the objective function.

$fit(\boldsymbol{x})$, the fitness value of $\boldsymbol{x}$, is calculated by (26):

$$fit(\boldsymbol{x})=\frac{1}{1+f(\boldsymbol{x})} \tag{26}$$

Initialization process is the first step of the proposed algorithm. In ABC algorithm, the initial swarm composed of employed bees are given by (1). But the lower bound and the upper bound of the $\boldsymbol{x}$ in sequence matching are difficult to acquire. Thus, the initialization method should be redefined.

The proposed algorithm applies scaling transformation, rotation transformation, and translation transformation on the INS-indicated sequence to implement initialization. Translation transformation is implemented by choosing a random first element of initial swarm:

$$x_{i,1}^{L}=x_{\min}^{L}+rand(0,1)(x_{\max}^{L}-x_{\min}^{L}) \tag{27}$$

$$x_{i,1}^{\lambda}=x_{\min}^{\lambda}+rand(0,1)(x_{\max}^{\lambda}-x_{\min}^{\lambda}) \tag{28}$$

Where $x_{i,1}^{L}$ and $x_{i,1}^{\lambda}$ indicate the latitude and longitude of the first sampling point on sequence i, $i=\{1,2,\cdots,SN\}$ and SN is the size of population. $x_{\max}^{L}$ and $x_{\min}^{L}$ are, respectively, the upper bound and the lower bound of latitude in search scope. $x_{\max}^{\lambda}$ and $x_{\min}^{\lambda}$ are, respectively, the upper bound and the lower bound of longitude in search scope.

Let S_i^{ini} be the scale factor used for initialization sequence i. The value of S_i^{ini} is calculated by (29):

$$S_i^{\text{ini}}=\begin{cases} rand(1,S), & S\geq 1\\ rand(S,1), & S<1\end{cases} \tag{29}$$

The expression of S is obtained from (24). The rotation angle used for initialization sequence i can be calculated by the following equation:

$$\beta_i^{\text{ini}}=rand(\beta_{\min}^{\text{ini}},\beta_{\max}^{\text{ini}}) \tag{30}$$

Where $\beta_{\min}^{\text{ini}}$ and $\beta_{\max}^{\text{ini}}$ are the minimum angle error and the maximum angle error of a segment in INS-indicated trajectory. Thus, the rotation matrix used for initialization sequence i have been derived:

$$\boldsymbol{R}_i^{\text{ini}}=\begin{bmatrix}\cos\beta_i^{\text{ini}} & -\sin\beta_i^{\text{ini}}\\ \sin\beta_i^{\text{ini}} & \cos\beta_i^{\text{ini}}\end{bmatrix} \tag{31}$$

Let $\boldsymbol{X}^{\text{INS}}=\left\{\boldsymbol{x}_1^{\text{INS}},\boldsymbol{x}_2^{\text{INS}},\ldots,\boldsymbol{x}_N^{\text{INS}}\right\}$ be the INS-indicated trajectory and $\boldsymbol{X}_i^{\text{ini}}=\left\{\boldsymbol{x}_{i,1}^{\text{ini}},\boldsymbol{x}_{i,2}^{\text{ini}},\ldots,\boldsymbol{x}_{i,N}^{\text{ini}}\right\}$ be the ith initialization sequence. After above mentioned transformations have been performed, the initial swarm is obtained:

$$\boldsymbol{X}_i^{\text{INS}\prime}=S_i^{\text{ini}}\boldsymbol{R}_i^{\text{ini}}\boldsymbol{X}^{\text{INS}} \tag{32}$$

$$\boldsymbol{T}_i^{\text{ini}}=\boldsymbol{x}_{i,1}^{\text{ini}}-\boldsymbol{x}_1^{\text{INS}\prime} \tag{33}$$

$$\boldsymbol{X}_i^{\text{ini}}=\boldsymbol{X}_i^{\text{INS}\prime}+\boldsymbol{T}_i^{\text{ini}} \tag{34}$$

Where $\boldsymbol{X}_i^{\text{INS}\prime}$ is the intermediate sequence, $\boldsymbol{x}_1^{\text{INS}\prime}$ is the first point in $\boldsymbol{X}_i^{\text{INS}\prime}$. Additional translation vector need to be applied on generated trajectories if they are out of range. Let $init_{\min}^{L}$ and $init_{\max}^{L}$ be the maximum and minimum latitude of initial trajectory. $init_{\min}^{\lambda}$ and $init_{\max}^{\lambda}$ are the maximum and minimum longitude of initial trajectory. $\mathbf{T}=\left\{T^{L},T^{\lambda}\right\}$, the translation vector, can be expressed as:

$$T^{L}=\begin{cases} x_{\min}^{L}-init_{\min}^{L}, & init_{\min}^{L}<x_{\min}^{L}\\ x_{\max}^{L}-init_{\max}^{L}, & init_{\max}^{L}>x_{\max}^{L}\\ 0, & \text{other}\end{cases} \tag{35}$$

$$T^{\lambda}=\begin{cases} x_{\min}^{\lambda}-init_{\min}^{\lambda}, & init_{\min}^{\lambda}<x_{\min}^{\lambda}\\ x_{\max}^{\lambda}-init_{\max}^{\lambda}, & init_{\max}^{\lambda}>x_{\max}^{\lambda}\\ 0, & \text{other}\end{cases} \tag{36}$$

Employed Bee Phase

With the initialization complete, an employed bee updates its food source in the neighborhood. Suppose $\boldsymbol{X}_i^{\text{ini}}$ is the sequence to be updated, $\boldsymbol{X}_j^{\text{ini}}=\left\{\boldsymbol{x}_{j,1}^{\text{ini}},\boldsymbol{x}_{j,2}^{\text{ini}},\ldots,\boldsymbol{x}_{j,N}^{\text{ini}}\right\}$ is selected randomly in the neighborhood of $\boldsymbol{X}_i^{\text{ini}}$, $j\in\{1,2,\cdots,i-1,i+1,\cdots,SN\}$ and SN is the size of population. The evolution equation is derived through the relation between $\boldsymbol{X}_i^{\text{ini}}$ and $\boldsymbol{X}_j^{\text{ini}}$.

Denoting with $\boldsymbol{Q}$ the covariance matrix:

$$\boldsymbol{Q}=\begin{bmatrix}Q_{11} & Q_{12}\\ Q_{21} & Q_{22}\end{bmatrix}=\sum_{k=1}^{N}\left(\boldsymbol{x}_{i,k}^{\text{ini}}-\tilde{\boldsymbol{x}}_i^{\text{ini}}\right)\left(\boldsymbol{x}_{j,k}^{\text{ini}}-\tilde{\boldsymbol{x}}_j^{\text{ini}}\right)^{\text{T}} \tag{37}$$

Where $\tilde{\boldsymbol{x}}_i^{\text{ini}}$ and $\tilde{\boldsymbol{x}}_j^{\text{ini}}$ are, respectively, the average value of all sampling points in $\boldsymbol{X}_i^{\text{ini}}$ and $\boldsymbol{X}_j^{\text{ini}}$. The eigenvalues of $\boldsymbol{Q}$ and the rotation angle from $\boldsymbol{X}_i^{\text{ini}}$ to $\boldsymbol{X}_j^{\text{ini}}$ are calculated through quaternion algorithm (Berthold, 1987):

$$\begin{aligned} \lambda_{1,2} &= \pm\left[(Q_{11}+Q_{22})^2+(Q_{21}-Q_{12})^2\right]^{1/2} \\ \lambda_{3,4} &= \pm\left[(Q_{11}-Q_{22})^2+(Q_{21}+Q_{12})^2\right]^{1/2} \end{aligned} \tag{38}$$

$$tg(\frac{\tau}{2}) = (Q_{11}+Q_{22}-\lambda_{\text{m}})/(Q_{12}-Q_{21}) \tag{39}$$

Where λ_μ (μ=$\{1,2,3,4\}$) are four eigenvalues of $\boldsymbol{Q}$. λ_{m} is the maximum eigenvalue. τ is the rotation angle from $\boldsymbol{X}_i^{\text{ini}}$ to $\boldsymbol{X}_j^{\text{ini}}$. Then, the rotation angle used for $\boldsymbol{X}_i^{\text{ini}}$ during employed bee phase is calculated by the following equations:

$$\beta_i^{\text{em}}\text{=}rand(-|\tau|,|\tau|) \tag{40}$$

Notice that the angle between $\boldsymbol{X}_i^{\text{ini}}$ and $\boldsymbol{X}^{\text{INS}}$ after rotation should be in the range $(\beta_{\min}^{\text{ini}}, \beta_{\max}^{\text{ini}})$. Denoting with η be the rotation angle from $\boldsymbol{X}_i^{\text{ini}}$ to $\boldsymbol{X}^{\text{INS}}$, β_i^{em} should meet the following in equation:

$$\beta_{\min}^{\text{ini}}+\eta < \beta_i^{\text{em}} < \beta_{\max}^{\text{ini}}+\eta \tag{41}$$

Then, the following expression is obtained by (40) and (41):

$$\beta_i^{\text{em}}\text{=}rand\left(\max(-|\tau|,\beta_{\min}^{\text{ini}}+\eta), min(|\tau|,\beta_{\max}^{\text{ini}}+\eta)\right) \tag{42}$$

Thus, the rotation matrix used for $\boldsymbol{X}_i^{\text{ini}}$ during employed bee phase is calculated by (43):

$$\boldsymbol{R}_i^{\text{em}}\text{=}\begin{bmatrix} \cos\beta_i^{\text{em}} & -\sin\beta_i^{\text{em}} \\ \sin\beta_i^{\text{em}} & \cos\beta_i^{\text{em}} \end{bmatrix} \tag{43}$$

Let L_i^{ini} and L_j^{ini} be the length of $\boldsymbol{X}_i^{\text{ini}}$ and $\boldsymbol{X}_j^{\text{ini}}$. L^{ins} is the length of INS-indicated trajectory. The scale factor used for $\boldsymbol{X}_i^{\text{ini}}$ during employed bee phase is calculated by the following equations:

$$S_i^{\text{em}} = rand(\min(\frac{L_j^{\text{ini}}}{L_i^{\text{ini}}},\frac{L_i^{\text{ini}}}{L_j^{\text{ini}}}), max(\frac{L_j^{\text{ini}}}{L_i^{\text{ini}}},\frac{L_i^{\text{ini}}}{L_j^{\text{ini}}})) \tag{44}$$

Notice that the scale factor between $\boldsymbol{X}_i^{\text{ini}}$ and $\boldsymbol{X}^{\text{INS}}$ after scaling transformation should meet the range on the right side of (29):

$$\begin{cases} 1\cdot L^{\text{ins}} < L_i^{\text{ini}}\cdot S_i^{\text{em}}<S\cdot L^{\text{ins}},\ S\geq 1 \\ S\cdot L^{\text{ins}} < L_i^{\text{ini}}\cdot S_i^{\text{em}}<1\cdot L^{\text{ins}},\ S<1 \end{cases} \tag{45}$$

Denoting with $S_{\min}^{\text{em}}$ and $S_{\max}^{\text{em}}$ the lower bound and the upper bound of S_i^{em}. According to (44) and (45), S_i^{em} is updated by (46):

$$S_i^{\text{em}} = rand\left(S_{\min}^{\text{em}}, S_{\text{m ax}}^{\text{em}}\right) \tag{46}$$

$$S_{\min}^{\text{em}} = \begin{cases} \max(min(\frac{L_j^{\text{ini}}}{L_i^{\text{ini}}},\frac{L_i^{\text{ini}}}{L_j^{\text{ini}}}),\frac{L^{\text{ins}}}{L_i^{\text{ini}}}), & S\geq 1 \\ \max(min(\frac{L_j^{\text{ini}}}{L_i^{\text{ini}}},\frac{L_i^{\text{ini}}}{L_j^{\text{ini}}}),S\cdot\frac{L^{\text{ins}}}{L_i^{\text{ini}}}), & S<1 \end{cases} \tag{47}$$

$$S_{\max}^{\text{em}} = \begin{cases} \min(\max(\frac{L_j^{\text{ini}}}{L_i^{\text{ini}}},\frac{L_i^{\text{ini}}}{L_j^{\text{ini}}}),S\cdot\frac{L^{\text{ins}}}{L_i^{\text{ini}}}), & S\geq 1 \\ \min(\max(\frac{L_j^{\text{ini}}}{L_i^{\text{ini}}},\frac{L_i^{\text{ini}}}{L_j^{\text{ini}}}),\cdot\frac{L^{\text{ins}}}{L_i^{\text{ini}}}), & S<1 \end{cases}$$

$\boldsymbol{X}_i^{\text{ini}}$ is converted to $\boldsymbol{X}_i^{\text{ini}\prime}$ after rotation transformation and scaling transformation have been performed:

$$\mathbf{X}_i^{\text{ini}\prime} = S_i^{\text{em}}\mathbf{R}_i^{\text{em}}\mathbf{X}_i^{\text{ini}} \tag{48}$$

Let $\tilde{\boldsymbol{x}}_i^{\text{ini}\prime}$ be the average value of all sampling points in $\boldsymbol{X}_i^{\text{ini}\prime}$ and the translation vector is calculated by (49):

$$\boldsymbol{T}_i^{\text{em}} = (\tilde{\boldsymbol{x}}_j^{\text{ini}} - \tilde{\boldsymbol{x}}_i^{\text{ini}\prime})\cdot rand(-1,1) \tag{49}$$

Then, the employed bees update their sources according to the following equation:

$$\boldsymbol{X}_i^{\text{em}} = \boldsymbol{X}_i^{\text{ini}\prime} + \boldsymbol{T}_i^{\text{em}} \tag{50}$$

If the fitness value of $\boldsymbol{X}_i^{\text{em}}$ is larger than the fitness value of $\boldsymbol{X}_i^{\text{ini}}$, $\boldsymbol{X}_i^{\text{ini}}$ will be replaced by $\boldsymbol{X}_i^{\text{em}}$. At the end of the employed bee phase, all of current sequences are denoted by $\boldsymbol{X}^{\text{em}}$.

Onlooker Bee Phase and Scout Bee Phase

The probability of each sequence to be selected by the onlooker bee is calculated by (4). Suppose $\boldsymbol{X}_i^{\text{em}}$ is the sequence selected by onlooker bee, the rotation matrix $\boldsymbol{R}_i^{\text{on}}$, the scale factor S_i^{on}, and the translation vector $\boldsymbol{T}_i^{\text{on}}$ used on $\boldsymbol{X}_i^{\text{em}}$ during onlooker bee phase can be calculated by (43), (46),and (49). Thus, the onlooker bees update their sources according to the following equation:

$$\boldsymbol{X}_i^{\text{on}} = S_i^{\text{on}}\boldsymbol{R}_i^{\text{on}}\boldsymbol{X}_i^{\text{em}} + \boldsymbol{T}_i^{\text{on}} \tag{51}$$

If the fitness value of $\boldsymbol{X}_i^{\text{on}}$ is larger than the fitness value of$\boldsymbol{X}_i^{\text{em}}$, $\boldsymbol{X}_i^{\text{em}}$ will be replaced by $\boldsymbol{X}_i^{\text{on}}$. The best solution achieved so far is memorized. In addition, a sequence will be abandoned when limit, a control parameter, is exceeded for the source during scout bee phase. So it is replaced with randomly produced solution by (34). The pseudo of the proposal is given below.

Procedure of the proposal

```
Begin
    1. confirm search scope
    2. initialize a matching sequence $\mathbf{X}_i^{\text{ini}}$ at generation $t = 0$
with $trial(i) = 0$ using (34) for $i = 1,2,\ldots,SN$
    3. evaluate $f\left(\mathbf{X}_i^{\text{ini}}\right)$ for $i = 1,2,\ldots,SN$ and set the best
solution as $\mathbf{X}_{\text{best}}$
    4. while termination condition is not reached do
//employed bee phase
        1) set $k = 1$, while $k<SN+1$ do
            Produce a new solution $\mathbf{X}_k^{\text{em}}$ for $\mathbf{X}_k^{\text{ini}}$ using (50)
            if $f\left(\mathbf{X}_k^{\text{ini}}\right) > f\left(\mathbf{X}_k^{\text{em}}\right)$, $\mathbf{X}_k^{\text{ini}} = \mathbf{X}_k^{\text{em}}$, trial$(i) = 0$
            else $trial(i) = trial(i)+1$
            end if
            $k = k+1$
            end while
        2) all of current sequences are denoted by $\mathbf{X}^{\text{em}}$
//onlooker bee phase
        3) set $k = 1$, while $k<SN+1$ do
        Select a sequence $\mathbf{X}_i^{em}$ based on its probability of
    selection calculated using (4) and produce a new solution
    $\mathbf{X}_i^{on}$ using (51)
        If $f\left(\mathbf{X}_i^{em}\right) > f\left(\mathbf{X}_i^{on}\right)$, $\mathbf{X}_i^{em} = \mathbf{X}_i^{on}$, $trial(i) = 0$
        else $trial(i) = trial(i)+1$
        end if
        $k = k+1$
        **end while**
      4) all of current sequences are denoted by $\mathbf{X}^{on}$
    //**scout bee phase**
      5) denote the best solution in $\mathbf{X}^{on}$ as $\mathbf{X}_{best}{}'$
        if $f\left(\mathbf{X}_{best}\right) > f\left(\mathbf{X}_{best}{}'\right)$, $\mathbf{X}_{best} = \mathbf{X}_{best}{}'$
        end if
      6) reinitialize $i$th sequence if $trial(i) > limit$
      7) $t = t+1$
      8) all of current sequences are denoted by $\mathbf{X}^{ini}$
    **end while**
    5. output $\mathbf{X}_{best}$
    **End**
```

TABLE 1 | Simulation condition of INS.

Parameters	Quantity	Unit
Gyro constant drift	0.02	°/h
Gyro random drift (1σ)	0.02	°/h
Accelerometer constant bias	100	μg
Accelerometer random bias (1σ)	100	μg
Velocity	7.71	m/s
Acceleration	0	m/s
Initial angle error	0	°
Azimuth angle	60	°
Initial longitude error	0.1	′
Initial latitude error	0.1	′
Simulation time	16	h

SIMULATION AND ANALYSIS

Simulation Parameters

In order to test the feasibility of our approach, numerous simulation experiments have been performed. An INS indicated trajectory start from (156°E, 20°N) is generated and the INS relevant parameters are listed in **Table 1**. The INS position error is shown in **Figure 1** (**Supplementary Data Sheet 1**). As seen from **Figure 1**, the position error of the generated trajectory exhibits a Scuhler oscillation of 84.4 min.

In order to evaluate the algorithm presented in this work, a gravity anomaly base map is required. In the simulations tests, goco05c model (Fecher et al., 2017) is used to calculate the gravity anomaly in the area from (147.6°E, 19.5°N) to (156.6°E, 24°N). After interpolated, the grid step is converted to 0.3′. The 3-D map of the gravity anomaly data is shown in **Figure 2** (**Supplementary Data Sheet 1**). The gravity anomaly relevant parameters are shown in **Table 2**.

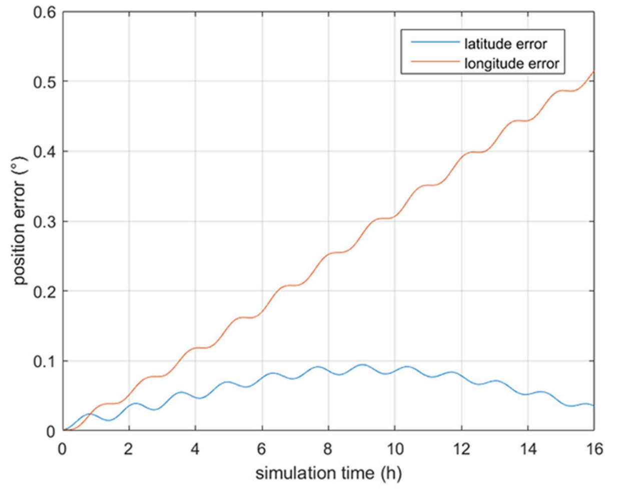

FIGURE 1 | The INS position error.

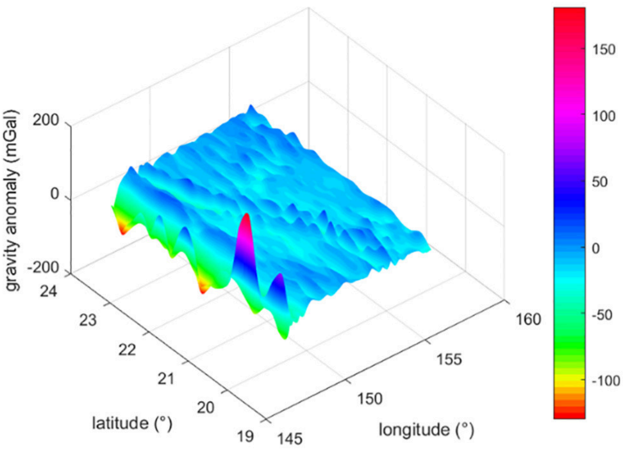

FIGURE 2 | 3-D gravity anomaly base map.

TABLE 2 | Parameters of gravity anomaly base map.

Model name	Goco05c
Number of grid points	900 × 1800
Grid step	0.3′
Minimum value	−130. 039 mGal
Maximum value	180. 902 mGal
Mean	−15.246 mGal

Figure 3 shows the angle error of a segment in INS-indicated trajectory. It is obtained that the angle error is vary in a small range and the maximum segment angle error is 15°. Thus, $\beta_{\min}^{ini}$ and $\beta_{m\,ax}^{ini}$ are set as −15°and 15°, respectively. The proposed algorithm uses the 3σ principle to confirm the search scope (Han et al., 2016). **Table 3** shows the configuration of the proposed algorithm. In this paper, optimal parameter values of affine transformation based ABC are obtained based on experience.

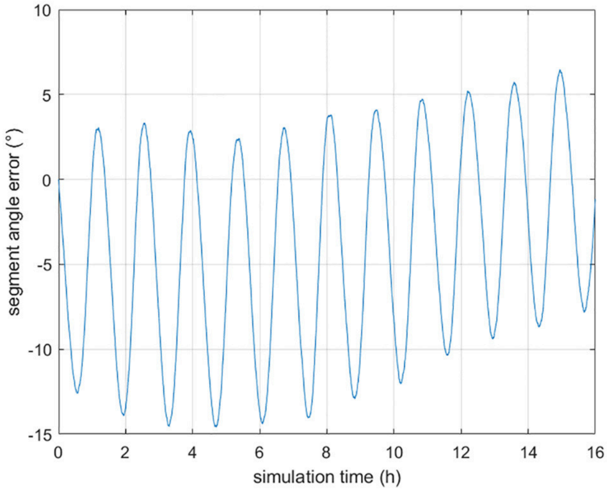

FIGURE 3 | The INS-indicated segment angle error.

TABLE 3 | Configuration of the proposed algorithm.

Maximum iterations	500
Limit	20
Population size	10
Number of sampling points per sequence	12
Sampling interval	5 min
The variance of gravity anomaly measurement noise	1 mGal

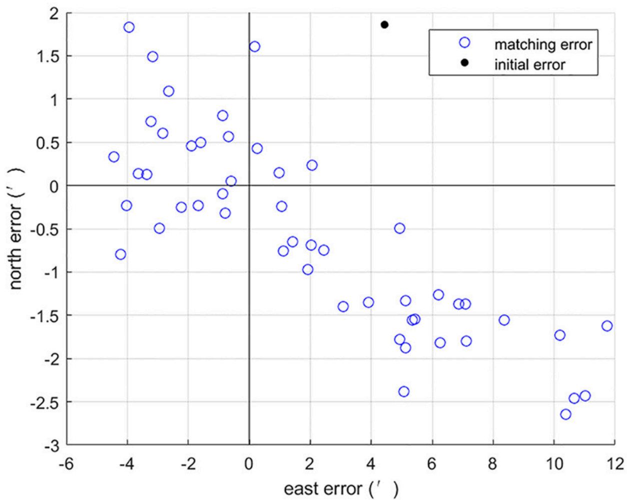

FIGURE 4 | Position errors of basic ABC.

Comparison Between Basic ABC and Affine Transformation Based ABC

Based on the parameters above, the first set of 50 Mont Carlo simulation tests were done on the trajectory after 3 h of sailing. Simulation results of introducing sequence matching strategy into ABC algorithm has better positioning precision and matching probability than the basic ABC, as shown in **Figures 4, 5**. Statistical results of simulation tests are given in **Table 4**. In **Table 4**, a matching result is considered to be a successful matching if the position error is within 2′.

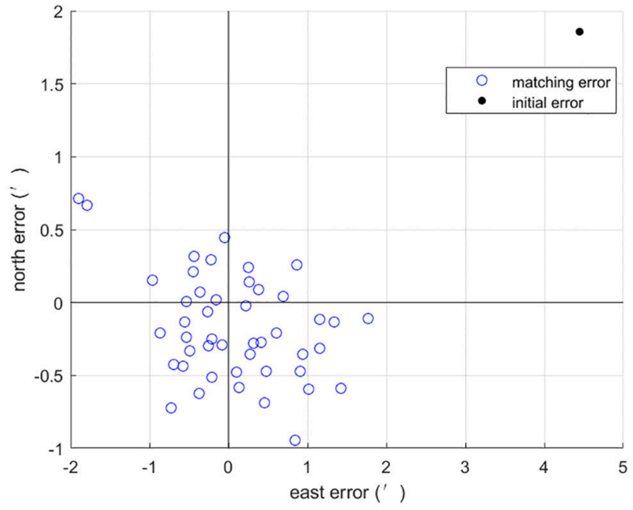

FIGURE 5 | Position errors of affine transformation based ABC.

TABLE 4 | Comparison of 50 Mont Carlo simulation results between basic ABC and affine transformation based ABC.

Algorithm	Initial error (′)	Mean error (′)	Matching probability (%)
Basic ABC	4.82	4.22	28
Affine transformation based ABC	4.82	0.73	96

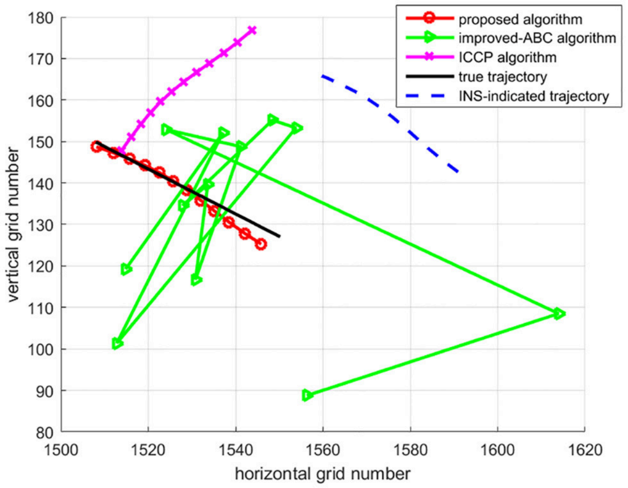

FIGURE 6 | Matched trajectory.

Comparison Between ICCP, Improved-ABC and Affine Transformation Based ABC

Due to ICCP is a widely used gravity matching algorithm and improved-ABC algorithm (Gao et al., 2014) is presented to tackle the similar problem. We use these algorithms as references, comparing the results of ICCP algorithm and improved-ABC algorithm with the proposed algorithm.

To test algorithms under large initial position errors, the second set of 50 Mont Carlo simulation tests were done on the trajectory after 7 h of sailing using the above three algorithms. Trajectories of matching results are shown in **Figure 6**, each trajectory is a typical one in 50 Mont Carlo simulations. **Figure 7** indicates the average error of longitude and latitude of each sampling point in 50 Mont Carlo simulations. In order to better evaluate the proposed algorithm, the statistical results of all the three algorithms are given in **Table 5**.

Simulation results in **Figures 6**, **7**, and **Table 5** show that the matching accuracy of the proposed algorithm is significantly superior to the other two algorithms under large initial position errors.

To better observe the effect of initial errors on these three algorithms, matching results of the whole trajectory within 16 h are shown in **Figure 8** and **Table 6**. The results are divided into three time periods for statistic. In **Table 6**, a matching

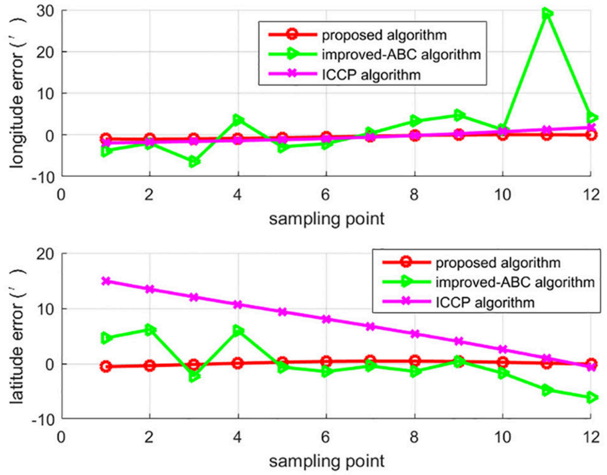

FIGURE 7 | The average error of longitude and latitude.

TABLE 5 | Statistical results of the matching errors.

Algorithm	Initial position error (km)	Mean error (km)	Standard deviation (km)
ICCP	22.47	25.85	0
Improved-ABC	22.47	15.95	8.70
Proposed algorithm	22.47	2.05	0.55

TABLE 6 | Statistical results within 16 h.

Time period	Algorithm	Average matching time (s)	Mean error (′)	Matching probability (%)
0–5 h	Improved-ABC	0.05	5.96	57
	ICCP	0.67	3.71	67
	Proposed algorithm	0.21	1.00	100
5–10 h	Improved-ABC	0.04	14.71	20
	ICCP	3.41	9.22	30
	Proposed algorithm	0.82	3.23	82
10–16 h	Improved-ABC	0.05	27.47	9
	ICCP	9.88	22.33	0
	Proposed algorithm	1.75	6.63	73

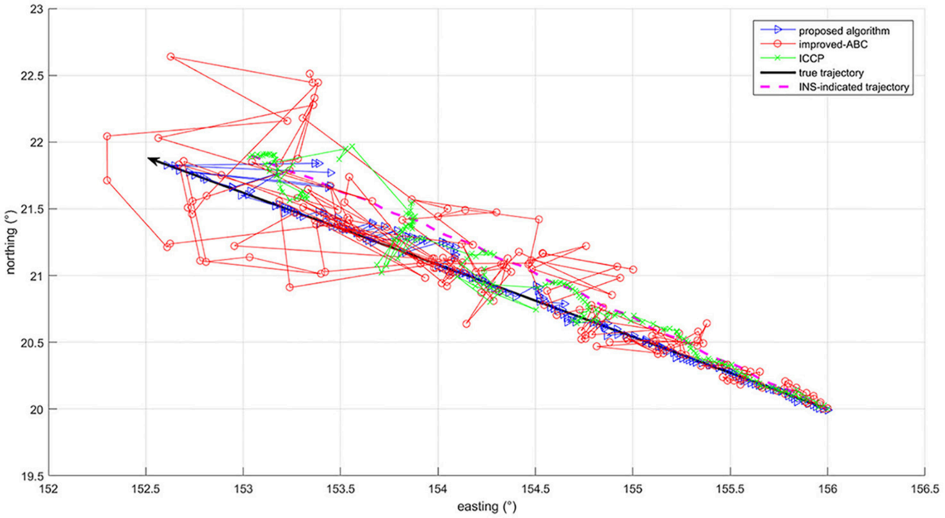

FIGURE 8 | Matching results of the whole trajectory.

result is considered to be a successful matching if the position error is within 5′. It can be concluded that the convergence speed of ICCP algorithm decreases a lot with the increase of initial position errors. While the convergence speed of improved-ABC algorithm is not affected by initial position errors. The proposed algorithm converge slower than improved-ABC algorithm and much faster than ICCP algorithm under large initial errors. In **Figure 8**, at the beginning, all of the three algorithms can hold high positioning accuracy. But with the initial errors and the search scope increase, the accuracy of ICCP algorithm and improved-ABC algorithm decrease significantly, whereas the proposed algorithm can guarantee much higher matching accuracy.

CONCLUSION

Gravity aided inertial navigation system can solve the problem of INS error accumulation and ensure the concealment of INS. However, there are still many problems in the field of gravity aided navigation system. In this paper, we focus on the problem of how to ensure the matching accuracy under the large initial errors condition. In order to avoid mismatching under large initial errors, an affine transformation based ABC algorithm is adopted. The simulation results show that the proposed algorithm can achieve high matching precision under the condition of large initial errors.

In addition to ABC algorithm, other evolutionary algorithms can also be used to solve the gravity matching problem. How to introduce the sequence matching strategy in other evolutionary algorithms? How are the effects of these algorithms after introducing the sequence matching strategy? These need to be further studied in the future. What's more, the matching accuracy of gravity matching algorithm is highly correlated with the precision and resolution of the gravity anomaly base map. This paper focuses on the matching algorithm for large initial errors. How to perform a gravity matching algorithm in the area with low uniqueness should be discussed in the next step.

AUTHOR CONTRIBUTIONS

TD and LM designed the study. TD, LM, and HS performed the simulations. All of the authors analyzed the data. All authors contributed to the writing and editing of the manuscript.

FUNDING

This work was supported by the National Natural Science Foundation of China (grant number 61473039).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnbot.2019.00019/full#supplementary-material

Supplementary Data Sheet 1 | Gravity base map and the INS data.

REFERENCES

Abrao, T. (2013). *Search Algorithms for Engineering Optimization: Chapter 8-Application of Harmony Search Algorithm in Power Engineering, Chapter Book-Chapter 8*. Crovatia: InTech Open Access Publishing.

Affleck, C. A., and Jircitano, A. (1990). "Passive gravity gradiometer navigation system," in *Proceedings of IEEE Position Location and Navigation Symposium* (Las Vegas, NV), 60–66.

Akay, B., and Karaboga, D. (2012). A modified artificial bee colony algorithm for real-parameter optimization. *Inform. Sci.* 192, 120–142. doi: 10.1016/j.ins.2010.07.015

Akbari, R., Hedayatzadeh, R., Ziarati, K., and Hassanizadeh, B. (2012). A multi-objective artificial bee colony algorithm. *Swarm Evol. Comput.* 2, 39–52. doi: 10.1016/j.swevo.2011.08.001

Allotta, B., Caiti, A., Costanzi, R., Fanelli, F., Fenucci, D., Meli, E., et al. (2016). A new AUV navigation system exploiting unscented Kalman filter. *Ocean Eng.* 113, 121–132. doi: 10.1016/j.oceaneng.2015.12.058

Berthold, K. P. H. (1987). Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A* 4, 629–642.

Bishop, G. C. (2002). Gravitational field maps and navigational errors [unmanned underwater vehicles]. *IEEE J. Ocean. Eng.* 27, 726–737. doi: 10.1109/JOE.2002.1040954

Canciani, A., and Raquet, J. (2017). Airborne magnetic anomaly navigation. *IEEE Trans. Aerosp. Electron. Syst.* 53, 67–80. doi: 10.1109/TAES.2017.2649238

Claus, B., and Bachmayer, R. (2015). Terrain-aided navigation for an underwater glider. *J. Field Robot.* 32, 935–951. doi: 10.1002/rob.21563

Copp, B., and Subbarao, K. (2015). Nonlinear adaptive filtering in terrain-referenced navigation. *IEEE Trans. Aerosp. Electron. Syst.* 51, 3461–3469. doi: 10.1109/TAES.2015.140826

Fecher, T., Pail, R., and Gruber, T. (2017). Goco05c: a new combined gravity field model based on full normal equations and regionally varying weighting. *Surv. Geophys*. 38, 571–590. doi: 10.1007/s10712-016-9406-y

Gao, W., Zhao, B., Zhou, G., Wang, Q., and Yu, C. (2014). Improved artificial bee colony algorithm based gravity matching navigation method. *Sensors* 14, 12968–12989. doi: 10.3390/s140712968

Han, Y., Wang, B., Deng, Z., and Fu, M. (2016). An improved TERCOM-based algorithm for gravity-aided navigation. *IEEE Sens. J.* 16, 2537–2544. doi: 10.1109/JSEN.2016.2518686

Hegrenaes, O., and Hallingstad, O. (2011). Model-aided INS with sea current estimation for robust underwater navigation. *IEEE J. Ocean. Eng.* 36, 316–337. doi: 10.1109/JOE.2010.2100470

Hidalgo-Paniagua, A., Vega-Rodríguez, M., and Ferruz, J. (2016). Applying the MOVNS (multi-objective variable neighborhood search) algorithm to solve the path planning problem in mobile robotics. *Expert Syst. Appl.* 58, 20–35. doi: 10.1016/j.eswa.2016.03.035

Kamgarparsi, B., and Kamgarparsi, B. (1999). Vehicle localization on gravity maps. *Proc. SPIE*. 3693, 182–191.

Karaboga, D., and Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J. Glob. Optimization* 39, 459–471. doi: 10.1007/s10898-007-9149-x

Kim, T., Kim, J., and Byun, S. W. (2018). A comparison of nonlinear filter algorithms for terrain-referenced underwater navigation. *Int. J. Control Autom. Syst.* 16, 2977–2989. doi: 10.1007/s12555-017-0504-5

Kim, Y., Hong, K., and Bang, H. (2018). Utilizing out-of-sequence measurement for ambiguous update in particle filtering. *IEEE Trans. Aerosp. Electron. Syst.* 54, 493–501. doi: 10.1109/TAES.2017.2741878

Lee, J., and Kim, D. W. (2016). An effective initialization method for genetic algorithm-based robot path planning using a directed acyclic graph. *Inf. Sci. (NY).* 332, 1–18. doi: 10.1016/j.ins.2015.11.004

Li, H., Liu, M., and Zhang, F. (2017). Geomagnetic navigation of autonomous underwater vehicle based on multi-objective evolutionary algorithm. *Front. Neurorobot.* 11, 1–8. doi: 10.3389/fnbot.2017.00034

Li, J., Pan, Q., and Duan, P. (2016). An improved artificial bee colony algorithm for solving hybrid flexible flowshop with dynamic operation skipping. *IEEE Trans. Cybern.* 46, 1311–1324. doi: 10.1109/TCYB.2015.2444383

Li, T., Zhao, D., and Huang, Z. (2013). A wavelet-based grey particle filter for self-estimating the trajectory of manoeuvring autonomous underwater vehicle. *Trans. Inst. Meas. Control* 36, 321–335. doi: 10.1177/0142331213500981

Paull, L., Saeedi, S., Seto, M., and Li, H. (2013). AUV navigation and localization: a review. *IEEE J. Ocean. Eng. 2013* 39, 131–149. doi: 10.1109/JOE.2013.2278891

Quan, W., and Fang, J. (2010). A star recognition method based on the adaptive ant colony algorithm for star sensors. *Sensors* 10, 1955–1966. doi: 10.3390/s100301955

Rice, H., Kelmenson, S., and Mendelsohn, L. (2004). "Geophysical navigation technologies and applications," in *Proceedings of IEEE Position Location and Navigation Symposium* (Monterey, CA), 618–624.

Sag, T., and Çunkaş, M. (2015). Color image segmentation based on multiobjective artificial bee colony optimization. *Appl. Soft Comput.* 34, 389–401. doi: 10.1016/j.asoc.2015.05.016

Simanek, J., Reinstein, M., and Kubelka, V. (2015). Evaluation of the EKF-based estimation architectures for data fusion in mobile robots. *IEEE ASME Trans. Mech.* 20, 985–990. doi: 10.1109/TMECH.2014.2311416

Song, Z., Zhang, J., Zhu, W., and Xi, X. (2016). The vector matching method in geomagnetic aiding navigation. *Sensors* 16:1120. doi: 10.3390/s16071120

Teymourian, E., Kayvanfar, V., Komaki, G. H. M., and Zandieh, M. (2016). Enhanced intelligent water drops and cuckoo search algorithms for solving the capacitated vehicle routing problem. *Inf. Sci. (NY).* 334, 354–378. doi: 10.1016/j.ins.2015.11.036

Tong, Y., Bian, S., Jiang, D., and Xiao, S. (2011). Gravity matching simulation of real-time iccp algorithm. *J. Chin. Inertial Technol.* 19, 340–343.

Wang, X., Su, M., Liu, P., and Ding, S. (2013). Application of improved ICCP algorithm in gravity matching aided navigation. *Sci. Surv. Mapp.* 207. 38, 36–39

Wu, L., Wang, H., Chai, H., Hsu, H., and Wang, Y. (2015). Research on the relative positions-constrained pattern matching method for underwater gravity-aided inertial navigation. *J. Navigation* 68, 937–950. doi: 10.1017/S0373463315000235

Wu, L., Wang, H., Chai, H., Zhang, L., Hsu, H., and Wang, Y. (2017). Performance evaluation and analysis for gravity matching aided navigation. *Sensors* 17:769. doi: 10.3390/s17040769

Xu, X., Wu, J., Xu, S., Wang, L., and Li, P. (2014). ICCP algorithm for underwater terrain matching navigation based on affine correction. *J. Chin. Inertial Technol.* 22, 362–367. doi: 10.13695/j.cnki.12-1222/o3.2014.03.016

Yan, G., Yan, W., and Xu, D. (2008). Application of simplified UKF in SINS initial alignment for large misalignment angles. *J. Chin. Inertial Technol.* 16, 253–264. doi: 10.13695/j.cnki.12-1222/o3.2008.03.006

Yildiz, A. R. (2013). Comparison of evolutionary-based optimization algorithms for structural design optimization. *Eng. Appl. Artif. Intell.* 26, 327–333. doi: 10.1016/j.engappai.2012.05.014

Zhao, L., Gao, N., Huang, B., Wang, Q., and Zhou, J. (2014). A novel terrain-aided navigation algorithm combined with the TERCOM algorithm and particle filter. *IEEE Sens. J.* 15, 1124–1131. doi: 10.1109/JSEN.2014.2360916

Zhu, Z., Guo, Y., and Yang, Z. (2015). Study on initial gravity map matching technique based on triangle constraint model. *J. Navigation* 69, 353–372. doi: 10.1017/S0373463315000661

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.