drones

MDPI

Article

Identification of INS Sensor Errors from Navigation Data Based on Improved Pigeon-Inspired Optimization

Zhihua Li [1,2,*], Yimin Deng [1] and Wenxue Liu [2]

[1] School of Automation Science and Electrical Engineering, Beihang University, Beijing 100083, China
[2] Beijing Aerospace Times Laser Inertial Technology Company, Ltd., Beijing 100094, China
* Correspondence: by1703133@buaa.edu.cn

Abstract: The error level of inertial sensor parameters determines the navigation accuracy of an inertial navigation system. For many applications, such as drones, errors in horizontal gyroscopes and accelerometers, can significantly affect the navigation results. Different from most methods of filter estimation, we innovatively propose using evolutionary algorithms, such as the improved pigeon-inspired optimization (PIO) method, to identify sensor errors through navigation data. In this method, the navigation data are firstly collected; then, the improved carrier pigeon optimization method is used to find the optimal error parameter values of the horizontal gyroscope and accelerometer, so as to minimize the navigation result error calculated by the navigation data. At the same time, we propose a new improved method for pigeon-inspired optimization with dimension vectors adaptive mutation (DVPIO for short) that can avoid local optima in the later stages of the iteration. In the DVPIO method, $2n$ particles with poor fitness are selected for the following variation, with $2n$ dimension vectors when it is judged that the position is premature, where n represents the number of parameters to be identified; a dimension vector only represents the positive or negative change of a parameter, whose change amount is d can be adjusted adaptively. DVPIO method has better stability, faster convergence speed, and higher accuracy. This work has potential to reduce the need for the disassembly and assembly of the INS and return it to the manufacturer for calibration.

Keywords: inertial navigation; sensor errors; pigeon-inspired optimization (PIO)

check for updates

Citation: Li, Z.; Deng, Y.; Liu, W. Identification of INS Sensor Errors from Navigation Data Based on Improved Pigeon-Inspired Optimization. *Drones* **2022**, *6*, 287. https://doi.org/10.3390/drones6100287

Academic Editor: Diego González-Aguilera

Received: 21 August 2022
Accepted: 28 September 2022
Published: 2 October 2022

1. Introduction

For the convenience of description, referring to relevant standards and terms, the acronyms and their full forms adopted in this article are shown in Table 1.

The inertial navigation system (INS for short) has the advantages of being completely autonomous, operating undisturbed, and providing the real-time output of various forms of navigation information, such as the attitude, position, and velocity, of the carrier. INS is widely used in various carriers, such as drones. The core components of INS are the gyroscope and accelerometer. A gyroscope is used to measure the angular velocity, and an accelerometer is used to measure acceleration. Generally, INS is composed of three gyroscopes and three accelerometers with an orthogonal distribution [1–5].

The autonomous navigation of a drone depends on INS and other various systems [6,7]. The combination of INS and the global navigation satellite system (GNSS for short) is the main navigation mode used by a drone at present [8]. When GNSS is degraded or stopped, high-precision INS autonomous navigation performance is very important. The INS of the drone can be combined with various sensors to improve the navigation performance. For example, Bassolillo et al. [9] studied the data fusion method of the INS.

The error sources of INS include component error, installation error, initial alignment error, navigation principle and method error, interference error, and external information error. The accuracy index of the INS gyroscope and accelerometer includes one-time power-on stability accuracy (noise) and multiple power-on stability accuracy (offset error). The noise and offset error of the gyroscope and accelerometer are important parts of the INS error, which affect the navigation accuracy of the system, and this influence will accumulate with the extension of the system operation time [10–12]. Sensor errors need to be calibrated before using the INS. Typical calibration methods used for sensor errors are the 12-position turnover method and the rotation method [13].

Table 1. Acronyms with their full forms.

Acronyms	Full Forms
INS	inertial navigation system
IMU	inertial navigation unit
MEMS	microelectromechanical system
GNSS	global navigation satellite system
EA	evolutionary algorithms
BSA	backtracking search optimization algorithm
GA	genetic algorithm
PSO	particle swarm optimization
PIO	pigeon-inspired optimization
DVPIO	pigeon-inspired optimization with dimension vectors adaptive mutation
LSTM	long-short-term memory neural network
UAV	unmanned aerial vehicle

Sensor errors also fluctuate over time. After a particular period time (such as a year), the INS needs to be disassembled from the drone, and then returned to the manufacturer for recalibration. The disassembly and calibration of the INS will greatly increase its use and maintenance costs. It would be of great significance if we could use the flight data of the drone to calibrate the errors of the INS. Scholars have studied some online calibration methods for drones. For example, Han et al. [14] executed the online calibration of MEMS inertial sensors, improving the estimation accuracy and robustness of the system navigation. Wang et al. [15] used other auxiliary sensors to calibrate the sensor errors during the use of the INS, while Xiao et al. [16] presented an online IMU self-calibration method for low-cost inertial sensors of the visual-inertial system. The traditional methods are generally based on the inertial navigation error state equation, and the observer design method can also be used for parameter estimation [17,18], but the algorithm is complex, and the motion trajectory and motion time of the inertial navigation system are strictly required to ensure objectivity.

Research on the application of intelligent methods of INS is relatively scarce. However, interest in this area has gradually increased in recent years: Chen et al. [19] combined deep learning technology with inertial navigation, improving the application effect of wearable navigation devices based on human walking; Wang et al. [20] designed an IMU calibration and noise suppression method based on wavelet transform, which applies an LSTM (long-short-term memory neural network); Weber et al. [21] studied how to use neural network filtering to improve the accuracy of IMU real-time attitude estimation.

The evolutionary algorithm simulates the cooperative behavior of animals and realizes collective wisdom beyond individual behavior. We often face the problem of how to break out of local optimization to obtain a higher accuracy during the use of evolutionary algorithms for optimization. Li et al. [22] proposed a novel inverse tangent chaotic inertia weight and sine learning factors and achieved a better convergence accuracy and speed. Zhang et al. [23] proposed a novel prey–predator PSO (PP-PSO) that employed the three strategies of catch, escape, and breeding and could achieve a better performance. Zaman et al. [24] proposed an improved PSO with BSA called PSOBSA to resolve the original PSO algorithm's problems; in this, the BSA's mutation and crossover operators were modified through the neighborhood to increase the convergence rate. In addition to that, a new mutation operator was introduced to improve the convergence accuracy and avoid falling into a local optimum in that paper.

Existing research mainly uses methods, such as Kalman filtering, to identify sensor errors through inertial navigation data and satellite navigation data. Evolutionary algorithm methods, such as PSO, can be used to identify optimal parameters but there are problems, such as easily falling into local optimum. PIO method has the advantages of fast convergence speed, but there are similar problems with PSO in the later iteration of the algorithm. Table 2 describes some of the limitations of the existing methods involved, which are also areas for improvement.

Table 2. Methods and limitations involved in this study.

Methods Involved	Limitations	References
Kalman filter estimation method to identify sensor errors through inertial navigation data and velocity errors during motion	The algorithm is complex, and the motion trajectory and the motion time of the inertial navigation system are strictly required;	[25,26]
PSO	It is easy to fall into local optimum, or it is difficult to further improve the accuracy in the later stage of iteration.	[27,28]
PIO	Although it has better convergence performance, it is easy to fall into local optimum, or it is difficult to further improve the accuracy in the later stage of iteration.	[29,30]

In order to reduce the need for the disassembly, assembly, and re-calibration of INS by the manufacturer, in this research, we studied how to use intelligent methods, such as pigeon-inspired optimization (PIO), to identify sensor errors through flight data according to the INS application requirements of the drone.

2. Method for Identifying Sensor Errors from Navigation Data Based on DVPIO

This section presents the methodology of this study. Firstly, the navigation principle of INS is expounded; then, the influence of the sensor error on the navigation results is analyzed; an improved PIO method (DVPIO method) is proposed; finally, combining the influence mechanism of sensor errors on navigation results and the DVPIO method, a method for identifying sensor errors from navigation data based on DVPIO is constructed.

For the convenience of description, referring to relevant standards and terms, the mathematical symbols and their descriptions are shown in Nomenclature.

2.1. Principles and Error Effects of Inertial Navigation

In this part, the function and composition of INS are firstly described. Then, the mechanism that INS performs inertial navigation calculation based on sensor data to provide speed, position, and attitude is expounded. Finally, the influence of the sensor error of the INS on the navigation results is analyzed, including the influence of the accelerometer error on the speed, the influence of the gyroscope error on the attitude and so on.

2.1.1. Principles of Inertial Navigation

We take the INS used by a drone as an example to illustrate the method, display the results, and explain the significance of this research for the engineering application of the drone platform. The INS shown in Figure 1, as the main navigation measurement equipment used in the drone platform, can independently provide the drone with attitude, velocity, position, acceleration, and angular rate information along the X, Y, and Z directions of the inertial measurement coordinate system (front, right, lower coordinate system). It can execute navigation solutions, provide navigation information, receive the satellite navigation data sent by the drone flight controlling system, and calculate the INS/GNSS integrated navigation results. The INS shown in Figure 1 is mainly composed of the following functional components: three optical gyroscopes, three accelerometers, a power supply system, a data acquisition system, a computer system, and a structural system.

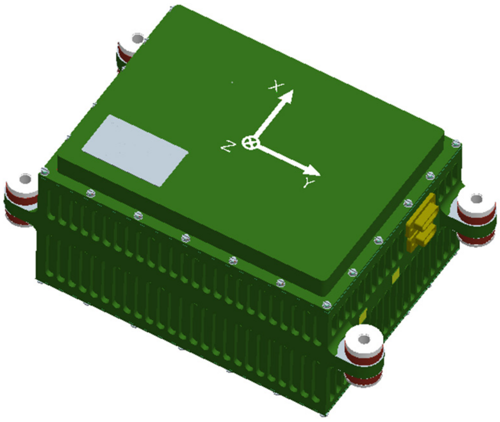


Figure 1. Optical INS.

The output pulse data of the gyroscopes and accelerometers of the INS are first converted and calculated using the calibration factor, zero bias, installation error, and other parameters. Then, the navigation solution is executed, before finally the position, velocity, and attitude information of the carrier is output. The navigation calculation of the geographic coordinate system (*l*-frame for short) of drones includes the attitude solution, coordinate transformation of the *l*-frame, navigation differential equation solution, and other processes.

The navigation differential equations of the *l*-frame are:

$$\begin{bmatrix} \dot{r}^{l} \\ \dot{V}^{l} \\ \dot{R}^{l} \end{bmatrix} = \begin{bmatrix} D^{-1}V^{l} \\ R_{b}^{l}f^{b} + g^{l} - \left(2\Omega_{ie}^{l} + \Omega_{el}^{l}\right)V^{l} \\ R_{b}^{l}\left(\Omega_{ib}^{b} - \Omega_{il}^{b}\right) \end{bmatrix} \tag{1}$$

$$D^{-1} = \begin{bmatrix} 0 & \frac{1}{R_M+h} & 0 \\ \frac{1}{(R_N+h)\cos\varphi} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

where *l* represents the *l*-frame, *b* represents the vehicle coordinate system (*b*-frame for short); *i* represents the inertial reference frame (*i*-frame for short); *e* represents the Earth coordinate system (*e*-frame for short); r^{l} is the latitude, longitude, and altitude of the *l*-frame; R_{b}^{l} is the conversion matrix from the *b*-frame to the *l*-frame; f^{b} is the specific force of the *b*-frame; g^{l} is the gravitational acceleration of the *l*-frame; Ω_{el}^{l} is an antisymmetric matrix of the angular rate vector expressed in the *l*-frame, where the angular rate vector is a vector of the *l*-frame relative to the *e*-frame; Ω_{ie}^{l}, Ω_{ib}^{b}, and Ω_{il}^{b} have similar meanings to Ω_{el}^{l}; *h* is the height; φ is the latitude; R_M is the radius of the meridional section; and R_N is the radius of the curvature in the prime vertica.

The differential equations of the quaternion of rotation are:

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ -\omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{3}$$

where ω represents ω_{lb}^{b} and is an angular rate vector that is expressed in the *b*-frame, where the angular rate vector is a vector for the *l*-frame relative to the *b*-frame; ω_x, ω_y, and ω_z are the coordinate components of ω; $q = [q_0\ q_1\ q_2\ q_3]'$ is the quaternion of rotation; and the attitude can be obtained by solving the quaternion differential equation.

2.1.2. Error Effects of Inertial Navigation

The pure inertial navigation of the INS in the event of satellite failure is critical for drones. The altitude of the drone can be obtained by an altimeter, so pure inertial navigation mainly considers the velocity and position errors in the horizontal direction. The system error is mainly caused by the zero bias errors of the gyroscopes and accelerometers in the horizontal position, and the zero bias errors tend to fluctuate over time.

The effect of the accelerometer zero bias errors on the specific force is as follows:

$$\begin{bmatrix} \delta f_x \\ \delta f_y \\ \delta f_z \end{bmatrix} = \begin{bmatrix} 1 & -E_{axz} & -E_{axy} \\ -E_{ayz} & 1 & -E_{ayx} \\ -E_{azy} & -E_{azx} & 1 \end{bmatrix} \begin{bmatrix} 1/K_{ax} & 0 & 0 \\ 0 & 1/K_{ay} & 0 \\ 0 & 0 & 1/K_{az} \end{bmatrix} \begin{bmatrix} \delta A_{0x} \\ \delta A_{0y} \\ \delta A_{0z} \end{bmatrix} \tag{4}$$

where δf_x, δf_y, and δf_z are the carrier specific force errors; E_{axz}, E_{axy}, E_{ayz}, E_{ayx}, E_{azy}, and E_{azx} are the installation errors of the accelerometers; K_{ax}, K_{ay}, and K_{az} are the scale factors of the accelerometers; and δA_{0x}, δA_{0y}, and δA_{0z} are the zero bias errors of the accelerometers.

Without considering the high-order small error, the equation for the INS velocity errors caused by the carrier specific force errors is as follows:

$$\begin{bmatrix} \delta \dot{v}_E \\ \delta \dot{v}_N \\ \delta \dot{v}_U \end{bmatrix} = \begin{bmatrix} 0 & f_U & -f_N \\ -f_U & 0 & f_E \\ f_N & -f_E & 0 \end{bmatrix} \begin{bmatrix} \delta p \\ \delta r \\ \delta A \end{bmatrix} + \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} \delta f_x \\ \delta f_y \\ \delta f_z \end{bmatrix} \tag{5}$$

where f_E, f_N, and f_U are the specific forces in the east, north, and up directions, respectively; δv_E, δv_N, and δv_U are the velocity errors in the east, north, and up directions, respectively; δp, δr, and δA are the pitch angle error, roll angle error, and azimuth angle error, respectively; and $R_{ij}(i = 1,2,3, j = 1,2,3)$ is the constituent element of matrix R_b^l.

The angular rate errors of navigation solution caused by gyros zero bias errors are:

$$\begin{bmatrix} \delta \omega_x \\ \delta \omega_y \\ \delta \omega_z \end{bmatrix} = \begin{bmatrix} 1 & -E_{gxz} & -E_{gxy} \\ -E_{gyz} & 1 & -E_{gyx} \\ -E_{gzy} & -E_{gzx} & 1 \end{bmatrix} \begin{bmatrix} 1/K_{gx} & 0 & 0 \\ 0 & 1/K_{gy} & 0 \\ 0 & 0 & 1/K_{gz} \end{bmatrix} \begin{bmatrix} \delta D_{0x} \\ \delta D_{0y} \\ \delta D_{0z} \end{bmatrix} \tag{6}$$

where $\delta\omega_x$, $\delta\omega_y$, and $\delta\omega_z$ are the angular rate errors; E_{gxz}, E_{gxy}, E_{gyz}, E_{gyx}, E_{gzy}, and E_{gzx} are the installation errors of the gyroscopes; K_{gx}, K_{gy}, and K_{gz} are the scale factors of the gyroscopes; and δD_{0x}, δD_{0y}, and δD_{0z} are the zero bias errors of the gyroscopes.

Without considering the high-order small error, the equation for the INS attitude errors caused by the angular rate errors is as follows:

$$\begin{bmatrix} \delta \dot{p} \\ \delta \dot{r} \\ \delta \dot{A} \end{bmatrix} = \begin{bmatrix} 0 & -\frac{1}{R_M+h} & 0 \\ \frac{1}{R_N+h} & 0 & 0 \\ \frac{\tan\varphi}{R_N+h} & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta v_E \\ \delta v_N \\ \delta v_U \end{bmatrix} - \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} \delta \omega_x \\ \delta \omega_y \\ \delta \omega_z \end{bmatrix} \tag{7}$$

2.2. DVPIO Method

Pigeon-inspired optimization (PIO) is a novel bio-inspired swarm intelligence optimization algorithm that was developed by Duan [31] in 2014. The PIO algorithm and its improved algorithms have been applied to different optimization issues, such as unmanned aerial vehicle (UAV) formation [32–34]. In this algorithm, by imitating the mechanism by which pigeons choose to use different navigational tools at different stages of finding a target, two different operators are proposed: a map and compass operator and a landmark operator.

Algorithm model of PIO:

$$\begin{gathered}\mathrm{X_i} = [\mathrm{x_{i1}}, \mathrm{x_{i2}}, \ldots \mathrm{x_{iD}}] \\ \mathrm{V_i} = [\mathrm{v_{i1}}, \mathrm{v_{i2}}, \ldots \mathrm{v_{iD}}] \\ \mathrm{V_i^{N_c}} = \mathrm{V_i^{N_{c-1}}} \mathrm{e}^{-\mathrm{R}\cdot\mathrm{N_c}} + \mathrm{rand}\left(\mathrm{X_{gbest}} - \mathrm{X_i^{N_{c-1}}}\right)\end{gathered} \tag{8}$$

$$\mathrm{X_i^{N_c}} = \mathrm{X_i^{N_{c-1}}} + \mathrm{V_i^{N_c}} \tag{9}$$

where $\mathrm{X_i} = [\mathrm{x_{i1}}, \mathrm{x_{i2}}, \ldots \mathrm{x_{iD}}]$ is the position and $\mathrm{V_i} = [\mathrm{v_{i1}}, \mathrm{v_{i2}}, \ldots \mathrm{v_{iD}}]$ is the velocity. The PIO algorithm updates $\mathrm{V_i}$ and $\mathrm{X_i}$ according to Equations (11) and (12). R is the map and compass factor, and the range of values is set from 0 to 1; rand is a random number with a range of values from 0 to 1; $\mathrm{N_c}$ is the current number of iterations; and $\mathrm{X_{gbest}}$ is the global optimal position obtained by comparing the positions of all pigeons after $\mathrm{N_{c-1}}$ iteration cycles.

The comparison of several studies shows that PIO has better global optimization performance and faster convergence speed than comparable algorithms [31,35].

It is not easy to avoid the local optimum in the later stage, since the cluster optimization algorithm undergoes random mutation in multiple dimensions simultaneously. Take a four-dimensional evolution algorithm as an example: Assuming that the optimal position of theory is [0, 0, 0, 0], a group of better positions [0.5, 0, 0, 0] can be found in the late iteration, and its fitness value is 0.2. After that, the position randomly generated by a particle swarm is in a form similar to [0.2, 0.2, −0.1, −0.2], with a fitness value of 0.5. Position [0.2, 0.2, −0.1, −0.2] is compared with [0.5, 0, 0, 0]; although the precision of the first dimension value is improved, the precision of the last three dimensions becomes worse, and the particle swarm is likely to eventually converge to [0.5, 0, 0, 0], so it is difficult to obtain further optimization.

A method for premature judgment and the self-adaptive mutation of dimension vectors based on their position is proposed to solve the above problems, which is as follows:

1. Premature judgment based on position:

The judgment value of whether the position is converged is:

$$\mathrm{Dist}^{\mathrm{N_c}} = \frac{\sum_{i=1}^{m} \sqrt{\sum_{j=1}^{D} \left(\mathrm{X_{gbestj}} - \mathrm{X}_{ij}^{\mathrm{N_c}}\right)^2}}{m} \tag{10}$$

where $\mathrm{X_{gbestj}}$ is the *j*th dimension of the global optimal position; $\mathrm{X}_{ij}^{\mathrm{N_c}}$ is the jth dimension position of the *i*th particle; *m* is the number of particles; and *D* is the number of dimensions. It can be considered that the particle swarm may appear precocious when $\mathrm{Dist}^{\mathrm{N_c}}$ is less than the threshold $\mathrm{Dist_{threshold}}$.

2. Dimension vectors mutations:

Taking the four-dimensional evolution algorithm as an example, eight dimension vectors can be defined as follows:

$$\begin{cases} V_1 = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix} \times d_{1p} \\ V_2 = \begin{bmatrix}-1 & 0 & 0 & 0\end{bmatrix} \times d_{1n} \\ V_3 = \begin{bmatrix}0 & 1 & 0 & 0\end{bmatrix} \times d_{2p} \\ V_4 = \begin{bmatrix}0 & -1 & 0 & 0\end{bmatrix} \times d_{2n} \\ V_5 = \begin{bmatrix}0 & 0 & 1 & 0\end{bmatrix} \times d_{3p} \\ V_6 = \begin{bmatrix}0 & 0 & -1 & 0\end{bmatrix} \times d_{3n} \\ V_7 = \begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix} \times d_{4p} \\ V_8 = \begin{bmatrix}0 & 0 & 0 & -1\end{bmatrix} \times d_{4n} \end{cases} \tag{11}$$

where d $= \begin{bmatrix} d_{1p} & d_{2p} & d_{3p} & d_{4p} \\ d_{1n} & d_{2n} & d_{3n} & d_{4n} \end{bmatrix}$ is the parameter of variation size; $d_{jp},\ d_{jn}(i = 1,2,3,4) > 0$ and can be adjusted adaptively. Eight particles with poor fitness are selected for the following dimension vector variation when it is judged that the position is premature:

$$\mathrm{X}_i^{\mathrm{N_c}} = \mathrm{X}_{\mathrm{gbest}} + V_i(i = 1 \sim 8) \tag{12}$$

Other particle positions are updated according to Formula (9).

3. Adaptive adjustment of mutations parameter *d*:

The fitness flag $\mathrm{T} = [\mathrm{T}_{\mathrm{dim1}}, \mathrm{T}_{\mathrm{dim2}}, \mathrm{T}_{\mathrm{dim3}}, \mathrm{T}_{\mathrm{dim4}}]$ of the dimension vector is introduced. This part takes the first dimension as an example for description, and other dimensions are similar to the first dimension. We assume that the dimension vector mutated according to Formula (12) is $\mathrm{X}_1^{\mathrm{N_c}}$, $\mathrm{X}_2^{\mathrm{N_c}}$, and the fitness flag of the minimum optimization problem is $\mathrm{T}_{\mathrm{dim1}}$:

$$\mathrm{T}_{\mathrm{dim1}}(\mathrm{i}) = \begin{cases} -1,\ if\ \mathrm{fitness}_{\mathrm{dim1n}} \leq \mathrm{fitness}_{\mathrm{dim1p}} \\ 1\ ,\ if\ \mathrm{fitness}_{\mathrm{dim1n}} > \mathrm{fitness}_{\mathrm{dim1p}} \end{cases} \tag{13}$$

where *i* is the evolutionary algebra; $\mathrm{fitness}_{\mathrm{dim1p}}$ is the fitness value of $\mathrm{X}_1^{\mathrm{N_c}}$; and $\mathrm{fitness}_{\mathrm{dim1n}}$ is the fitness value of $\mathrm{X}_2^{\mathrm{N_c}}$. The adaptive adjustment strategy of the variation *d* is:

$$\begin{cases} \begin{cases} d_{1p}(i) = 0.5d_{1n}(i-1) \\ d_{1n}(i) = 2d_{1n}(i-1) \end{cases}, & if{:}\mathrm{T}_{\mathrm{dim\,1}}(\mathrm{i}-2) = -1, \\ \begin{cases} d_{1p}(i) = 0.5d_{1p}(i-1) \\ d_{1n}(i) = 0.5d_{1p}(i-1) \end{cases}, & if : \mathrm{T}_{\mathrm{dim\,1}}(\mathrm{i}-2) = -1, \\ \begin{cases} d_{1p}(i) = 2d_{1p}(i-1) \\ d_{1n}(i) = 0.5d_{1p}(i-1) \end{cases}, & if : \mathrm{T}_{\mathrm{dim\,1}}(\mathrm{i}-2) = 1, \\ \begin{cases} d_{1p}(i) = 0.5d_{1n}(i-1) \\ d_{1n}(i) = 0.5d_{1n}(i-1) \end{cases}, & if{:}\mathrm{T}_{\mathrm{dim\,1}}(\mathrm{i}-2) = 1, \end{cases} \tag{14}$$

2.3. Identification of INS Sensor Errors Based on DVPIO

Based on the above analysis, the sensor errors of INS can be identified by the DVPIO method according to the navigation data of drone flights. Navigation data include sensor pulse data and satellite data, which can be collected and stored by the data acquisition instrument. The parameters of particle swarm optimization are the zero bias of the gyroscope and accelerometer in the horizontal position. The fitness function is the comprehensive statistical value of the velocity error of pure inertial navigation, in which the velocity error of pure inertial navigation is the difference between the pure inertial navigation velocity and the satellite velocity:

$$\begin{gathered} fitness = \textstyle\sum_{\mathrm{t}=1}^{n} dv_t \\ dv_t = \sqrt{\delta V_{Et}^2 + \delta V_{Nt}^2} \end{gathered} \tag{15}$$

where *t* is the time point; dv_t is the inertial navigation velocity error of INS at time *t*; δv_{Et} is the velocity error in the east at time *t*; δv_{Nt} is the velocity error in the north at time *t*; and *fitness* reflects the statistical value of the position error. The objective is to minimise the fitness function.

The steps of the flow chart are:

1. The initial value of particle swarm is generated randomly, and the fitness function is calculated to find the global optimum.
2. Calculation process of the fitness function:
   - The sensor data collected during the flight compensate for errors according to the position data of the particle swarm;

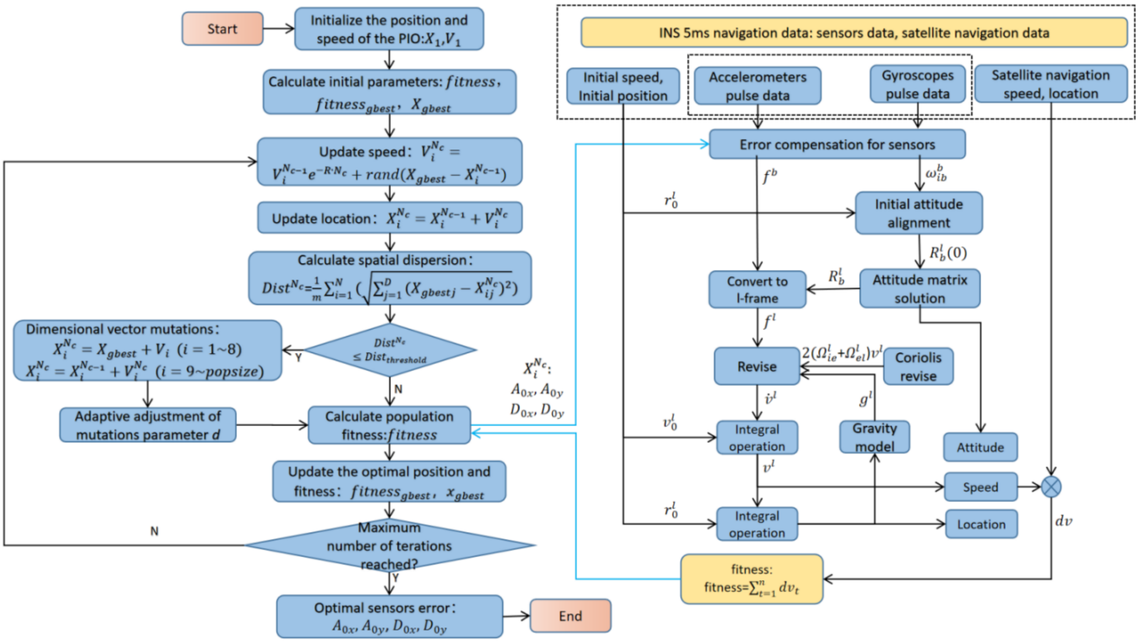
   - Obtain the initial parameters of inertial navigation: the initial velocity and position are obtained from the collected satellite navigation data, and the initial attitude is obtained from the initial alignment calculation;
   - Execute inertial navigation solution and obtain the difference between the velocity calculated, $\mathrm{v_i}$, and the satellite navigation velocity at the time, *i*, which is the velocity error, $\mathrm{dv_I}$;
   - Calculate the fitness function according to Formula (15).

3. Update the particle swarm velocity and position according to Formulas (8) and (9), and recalculate the fitness function to update the global optimization.
4. Judge whether the particle swarm has started prematurely according to Formula (10); if not, continue to execute (3). If it has started prematurely, execute (5).
5. The particle swarm is sorted according to its fitness, and natural selection is carried out. The eight particles with the largest fitness are randomly mutated according to Formula (12), and the number of mutations is self-adaptively adjusted according to Formulas (13) and (14).
6. Judge whether the convergence condition is reached. If so, the optimization calculation ends. If not, go to step 5 to continue the optimization calculation.

3. Results

3.1. Error Effects Results of Inertial Navigation

The INS sensor data are calculated according to Formulas (1)–(3), and the flow shown in Figure 2. The main parameters of inertial navigation calculation are shown in Table 3. When the zero bias value of the accelerometer and the zero bias value of the gyro are accurate, the result of the navigation solution is good, the inertial navigation velocity error is as shown in Figure 3, and the fitness value is calculated according to Formula (15) is 1.0×10^3 When the accelerometer X zero bias, the accelerometer Y zero bias, the gyro X zero bias, and the gyro Y zero bias have errors $\mathrm{\delta A_{0x}} = -2\sigma_\mathrm{A}$, $\mathrm{\delta A_{0y}} = 1.5\sigma_\mathrm{A}$, $\mathrm{\delta D_{0x}} = 2\sigma_\mathrm{D}$, and $\mathrm{\delta D_{0y}} = -1.5\sigma_\mathrm{D}$ ($\sigma_\mathrm{A} = 0.01°/\mathrm{h}$, $\sigma_\mathrm{D} = 0.0005\ \mathrm{m/s^2}$), respectively. Additionally, the inertial navigation velocity errors are larger, and their fitness values calculated by Formula (15) are 3.032×10^5, 1.959×10^5, 4.135×10^5, and 3.116×10^5, respectively. The results for these are shown in Figure 4, Figure 5, Figure 6, and Figure 7, respectively.



Start
Initialize the position and speed of the PIO:X_1,V_1
Calculate initial parameters: $fitness$, $fitness_{gbest}$, X_{gbest}
Update speed: $V_i^{N_c} = V_i^{N_c-1}e^{-R\cdot N_c} + rand(X_{gbest} - X_i^{N_c-1})$
Update location: $X_i^{N_c} = X_i^{N_c-1} + V_i^{N_c}$
Calculate spatial dispersion: $Dist^{N_c}=\frac{1}{m}\sum_{i=1}^{N}(\sqrt{\sum_{j=1}^{D}(X_{gbestj} - X_{ij}^{N_c})^2})$
$Dist^{N_c} \le Dist_{threshold}$
Y
N
Dimensional vector mutations: $X_i^{N_c} = X_{gbest} + V_i$ $(i = 1\sim8)$ $X_i^{N_c} = X_i^{N_c-1} + V_i^{N_c}$ $(i = 9\sim popsize)$
Adaptive adjustment of mutations parameter d
Calculate population fitness:$fitness$
$X_i^{N_c}$: A_{0x}, A_{0y} D_{0x}, D_{0y}
Update the optimal position and fitness: $fitness_{gbest}$, x_{gbest}
Maximum number of terations reached?
N
Y
Optimal sensors error: $A_{0x}, A_{0y}, D_{0x}, D_{0y}$
End
INS 5ms navigation data: sensors data, satellite navigation data
Initial speed, Initial position
Accelerometers pulse data
Gyroscopes pulse data
Satellite navigation speed, location
Error compensation for sensors
f^b
ω_{ib}^b
r_0^l
Initial attitude alignment
$R_b^l(0)$
Convert to l-frame
R_b^l
Attitude matrix solution
f^l
Revise
$2(\Omega_{ie}^l+\Omega_{el}^l)v^l$
Coriolis revise
$\dot{v}^l$
g^l
v_0^l
Integral operation
Gravity model
Attitude
v^l
Speed
r_0^l
Integral operation
Location
dv
fitness: fitness=$\sum_{t=1}^{n} dv_t$

Figure 2. Identification of INS sensor errors from navigation data based on DVPIO.

Table 3. Parameters for inertial navigation calculation.

Parameters	Value and Unit
Navigation interruption time	10 ms
Initial gravitational acceleration	9.801 m/s^2
Initial latitude	40.068°
Initial longitude	116.246
Initial height	50 m
Angular rate of the earth's rotation	7.292115147 × 10^5 rad/s
Long radius of the earth	6378245 m
Align time	10 min

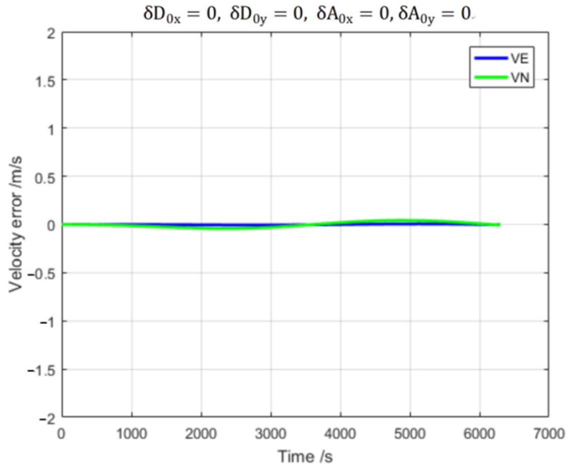


Figure 3. Velocity error when $\delta D_{0x} = 0$, $\delta D_{0y} = 0$, $\delta A_{0x} = 0$, and $\delta A_{0y} = 0$.

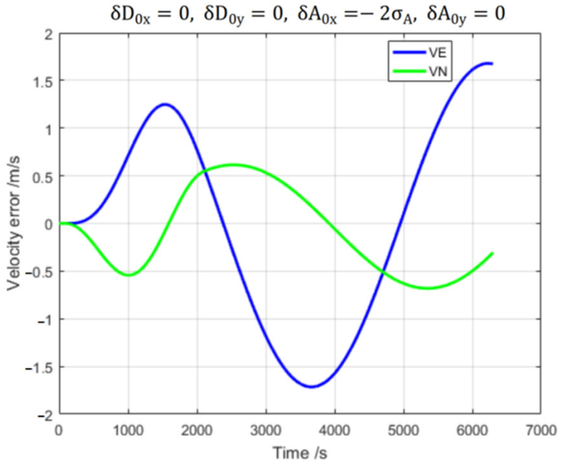


Figure 4. Velocity error when $\delta D_{0x} = 0$, $\delta D_{0y} = 0$, $\delta A_{0x} = -2\sigma_A$, and $\delta A_{0y} = 0$.

3.2. Identification Results of INS Sensor Errors Based on DVPIO

In most cases, the INS sensors' zero bias will change simultaneously over time. When the accelerometer X, accelerometer Y, gyro X, and gyro Y have zero bias errors of $\delta D_{0x} = 2\sigma_D$, $\delta D_{0y} = -1.5\sigma_D$, $\delta A_{0x} = -2\sigma_A$, and $\delta A_{0y} = 1.5\sigma_A$, respectively, the inertial navigation velocity error will be large, the fitness value will be 8.552 × 10^5, and the result will be as shown in Figure 8.

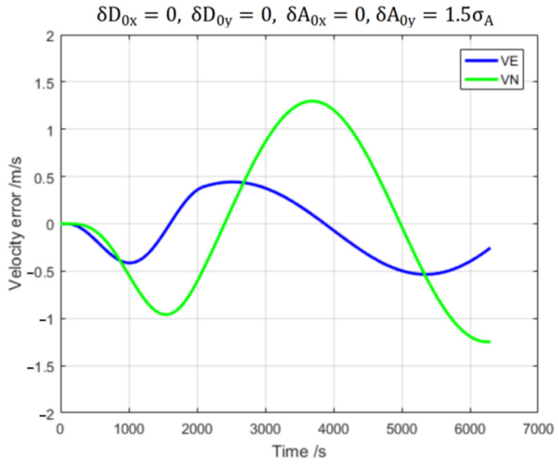


Figure 5. Velocity error when $\delta D_{0x} = 0$, $\delta D_{0y} = 0$, $\delta A_{0x} = 0$, and $\delta A_{0y} = 1.5\sigma_A$.

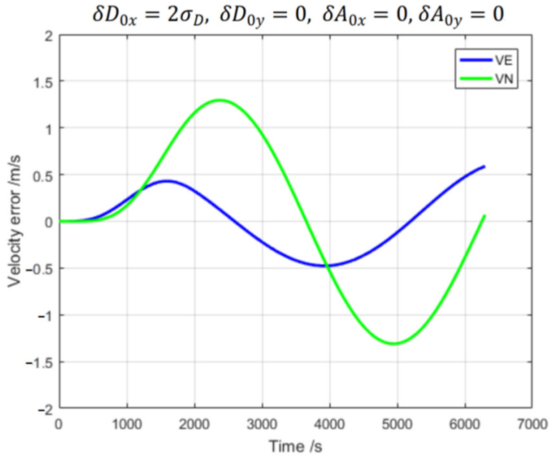


Figure 6. Velocity error when $\delta D_{0x} = 2\sigma_D$, $\delta D_{0y} = 0$, $\delta A_{0x} = 0$, and $\delta A_{0y} = 0$.

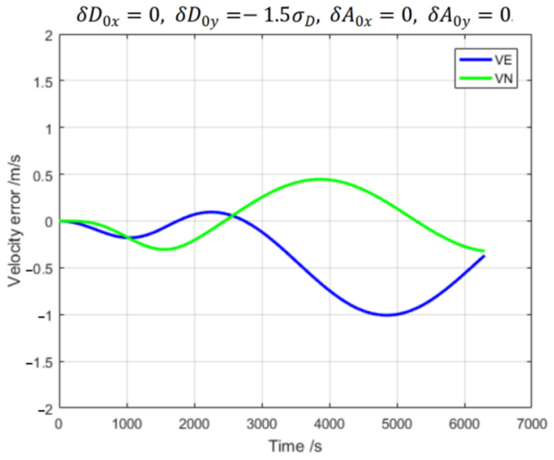


Figure 7. Velocity error when $\delta D_{0x} = 0$, $\delta D_{0y} = -1.5\sigma_D$, $\delta A_{0x} = 0$, and $\delta A_{0y} = 0$.

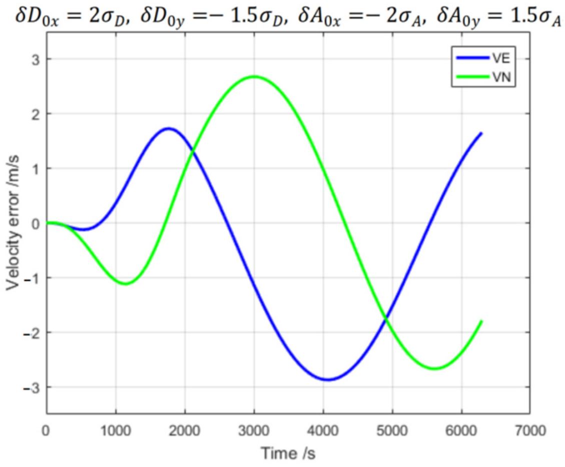


Figure 8. Velocity error when $\delta D_{0x} = 2\sigma_D$, $\delta D_{0y} = -1.5\sigma_D$, $\delta A_{0x} = -2\sigma_A$, and $\delta A_{0y} = 1.5\sigma_A$.

According to the process shown in Figure 2, GA, PSO, PIO, and DVPIO are used to identify the sensors zero bias errors δD_{0x}, δD_{0y}, δA_{0x}, and δA_{0y}. The parameters of the EA algorithm are shown in Table 4, and the results are shown in Figure 9.

Table 4. EA methods and parameters.

EA Methods	EA Parameters
GA	Number of particles: 20. Iteration steps: 40. Mating probability: 0.5. Mutation probability: 0.2.
PSO	Number of particles: 10. Iteration steps: 40. W = 0.8, C1 = 2, C2 = 2.
PIO	Number of particles: 10. Iteration steps: 40. R = 0.05, C = 1.
DVPIO	Number of particles: 10. Iteration steps: 40. R = 0.05, C = 1, d = 0.05 (initial value)

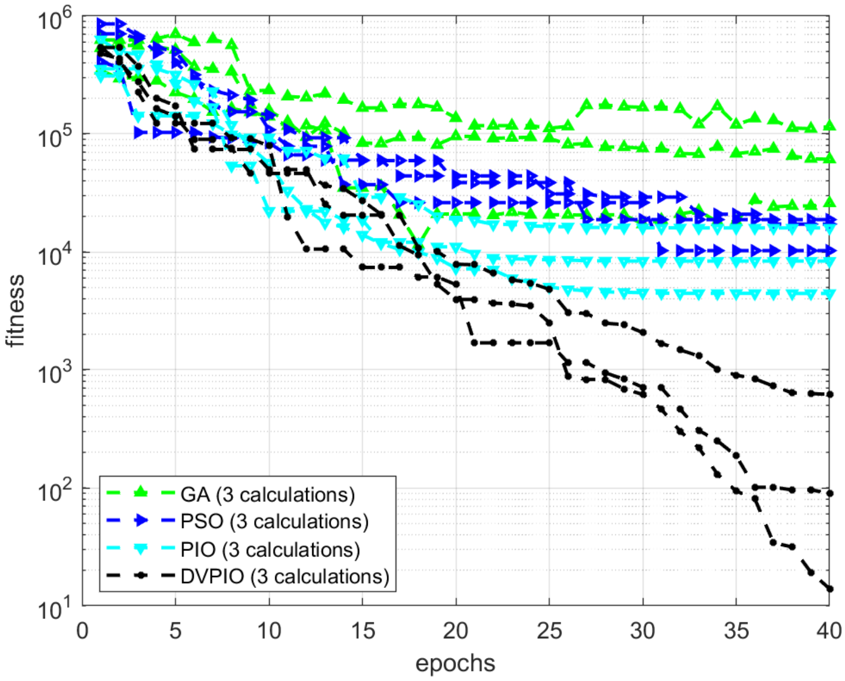


Figure 9. Fitness comparison of GA, PSO, PIO, and DVPIO (3 calculations for each method).

It can be seen from the comparison shown in Figure 9 that:

- The GA, PSO, PIO, and DVPIO methods can all be used to identify sensor errors from navigation data;
- Compared with the GA, PSO, and PIO methods, the PIO method has a faster convergence speed and a higher accuracy;
- The PIO method can easily fall into the local optimum in the later stage, while the DVPIO solves the problem of avoiding the local optimum and obtains more precise results.

3.3. Improve Analytics of DVPIO

During the PIO optimization shown in Figure 2, the position of each dimension of each particle is as shown in Figure 10, and the position of the global optimal ZBest is as shown in Figure 11. The particles of PIO begin to converge to [−1.92, 1.80, 2.03, −1.37] in the 20th generation, but the deviation of the 2nd dimension is still large at this time, and its deviation value is 0.3. It is difficult to avoid the vicinity of [−1.92, 1.80, 2.03, −1.37] in the later stage of PIO to obtain a better value.

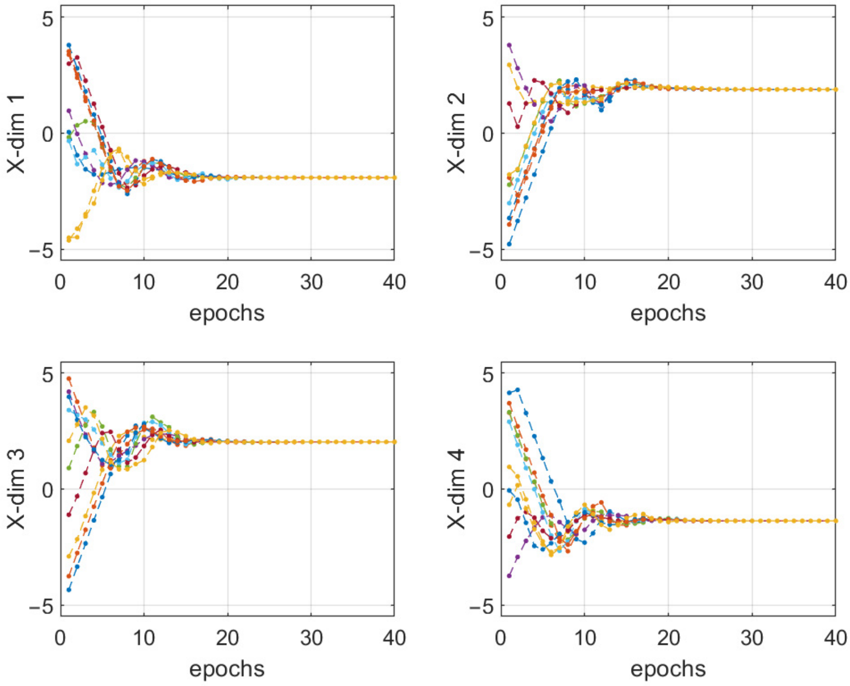


Figure 10. PIO particle swarm location distribution in each dimension(Different colors of lines represent different dimensions).

During the PIO optimization shown in Figure 2, the position of each dimension of each particle is as shown in Figure 12, and the position of the global optimal ZBest is as shown in Figure 13. The particle fitness of DVPIO continues to be optimized in the later stage. The calculation results show that the DVPIO method is more stable and has a higher accuracy. In the process of DVPIO optimization, a certain dimension vector, such as d_{2p}, is adaptively adjusted according to the change in fitness based on Formula (14). When the better dimension vectors of the previous two generations are in the same direction, the amount of variation is doubled; when the better dimension vectors of the previous two generations are in the opposite direction, this is considered to be close to the optimal position, and the amount of variation is reduced accordingly.

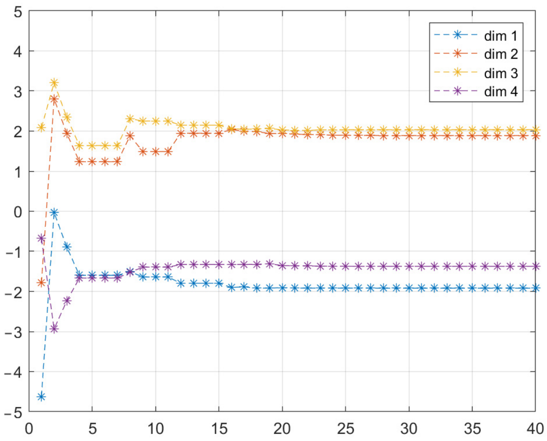


Figure 11. PIO particle swarm global optimal location.

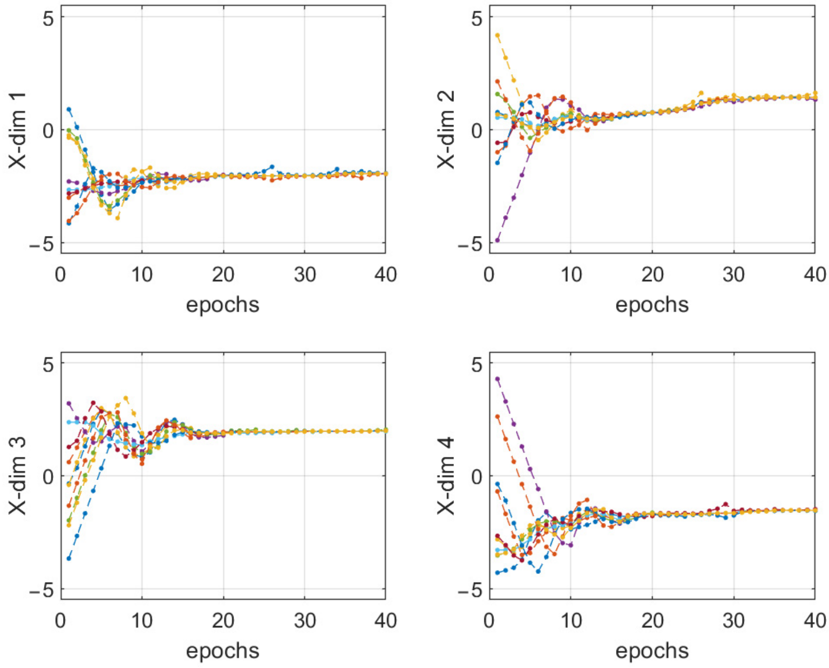


Figure 12. DVPIO particle swarm location distribution in each dimension(Different colors of lines represent different dimensions).

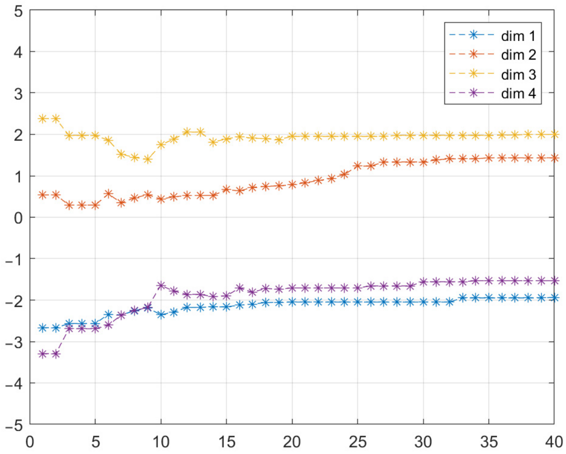


Figure 13. DVPIO particle swarm global optimal location.

4. Discussion

This paper used the evolutionary algorithms method to identify the sensor errors according to the flight data of the navigation system (including satellite navigation data and INS data) to realize the automatic calibration of the INS sensors on the drone. This method has potential to reduce the need for the disassembly and assembly of the INS and for it to be returned to the manufacturer for calibration. The results of several evolutionary algorithms show that the sensor errors can be accurately identified from the navigation data, and the pigeon-inspired optimization (PIO) has a higher accuracy and convergence speed. Aiming at the problem of evolutionary algorithms being likely to fall into local optimization in the later stage, this study analyzed the reasons for this and proposed a DVPIO method based on the self-adaptive mutation of dimension vectors on the basis of the PIO method. The calculation results show that the DVPIO method has a faster convergence speed and can converge to the optimal value more accurately in the later stage.

Author Contributions: Conceptualization, Methodology, Investigation, Software, Writing—original draft: Z.L.; Writing—review & editing, Funding acquisition, Project administration: Y.D.; Data curation, Resources: W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

l	represents *l*-frame, the geographic coordinate system
b	represents *b*-frame, the vehicle coordinate system
i	represents *i*-frame, the inertial reference frame
e	represents *e*-frame, the earth coordinate system
r^{l}	the vector of latitude, longitude, and altitude of the *l*-frame
$\mathrm{R}_{\mathrm{b}}^{\mathrm{l}}$	the conversion matrix from *b*-frame to *l*-frame; $\mathrm{R}_{\mathrm{ij}}(\mathrm{i}=1,2,3,\ \mathrm{j}=1,2,3)$ is its constituent element
$\Omega_{\mathrm{el}}^{\mathrm{l}}$	an antisymmetric matrix of the angular rate vector that is expressed in the *l*-frame, where the angular rate vector is a vector for the *l*-frame relative to the *e*-frame
f^{b}	the specific force of *b*-frame
v_{E}, v_{N}, v_{U}	the velocity in the east, north, and up directions, respectively
f_{E}, f_{N}, f_{U}	the specific forces in the east, north, and up directions, respectively
g^{l}	the gravitational acceleration of the *l*-frame
h, φ	the height and the latitude
R_{M}, R_{N}	the radius of meridional section and the radius of curvature in the prime vertica
$\omega(\omega_{\mathrm{lb}}^{\mathrm{b}})$	an angular rate vector expressed in the *b*-frame, where the angular rate vector is a vector for the *l*-frame relative to the *b*-frame
ω_{x}, ω_{y}, ω_{z}	the coordinate component of ω
$\mathrm{q}\ =[\mathrm{q}_0\ \mathrm{q}_1\ \mathrm{q}_2\ \mathrm{q}_3]'$	the quaternion of rotation
$\delta\mathrm{f}_{\mathrm{x}}$, $\delta\mathrm{f}_{\mathrm{y}}$, $\delta\mathrm{f}_{\mathrm{z}}$	the carrier specific force errors
$\delta\mathrm{v}_{\mathrm{E}}$, $\delta\mathrm{v}_{\mathrm{N}}$, $\delta\mathrm{v}_{\mathrm{U}}$	the velocity errors in the east, north, and up directions, respectively
$\delta\mathrm{p}$, $\delta\mathrm{r}$, $\delta\mathrm{A}$	the pitch angle error, roll angle error, and azimuth angle error, respectively
$\delta\omega_{\mathrm{x}}$, $\delta\omega_{\mathrm{y}}$, $\delta\omega_{\mathrm{z}}$	the angular rate errors
$\delta\mathrm{A}_{0\mathrm{x}}$, $\delta\mathrm{A}_{0\mathrm{y}}$, $\delta\mathrm{A}_{0\mathrm{z}}$	the zero bias errors of the accelerometers
$\delta\mathrm{D}_{0\mathrm{x}}$, $\delta\mathrm{D}_{0\mathrm{y}}$, $\delta\mathrm{D}_{0\mathrm{z}}$	the zero bias errors of the gyroscopes

$E_{axz}, E_{axy}, E_{ayz}, E_{ayx}, E_{azy}, E_{azx}$	the installation errors of the accelerometers, where E_{axz} means the installation error of the z accelerometer relative to the x axis
$E_{gxz}, E_{gxy}, E_{gyz}, E_{gyx}, E_{gzy}, E_{gzx}$	the installation errors of the gyroscopes, where E_{gxz} means the installation error of the z gyroscope relative to the x axis
K_{ax}, K_{ay}, K_{az}	the scale factors of the accelerometers
K_{gx}, K_{gy}, K_{gz}	the scale factors of the gyroscopes
$X_i = [x_{i1}, x_{i2}, \ldots x_{iD}]$	the position of the particle swarm
$V_i = [v_{i1}, v_{i2}, \ldots v_{iD}]$	the velocity of the particle swarm
N_c	the number of iterations
X_{gbest}	the global optimal position of the particle swarm

References

1. Kim, Y.; An, J.; Lee, J. Robust navigational system for a transporter using GPS/INS fusion. *IEEE Trans. Ind. Electron.* **2017**, *65*, 3346–3354. [CrossRef]
2. Gao, P.; Li, K.; Song, T. An accelerometers-size-effect self-calibration method for triaxis rotational inertial navigation system. *IEEE Trans. Ind. Electron.* **2017**, *65*, 1655–1664. [CrossRef]
3. Lee, J.; Lim, J.; Lee, J. Compensated heading angles for outdoor mobile robots in magnetically disturbed environment. *IEEE Trans. Ind. Electron.* **2017**, *65*, 1408–1419. [CrossRef]
4. Poddar, S.; Kumar, V.; Kumar, A. A comprehensive overview of inertial sensor calibration techniques. *J. Dyn. Syst. Meas. Control* **2017**, *139*, 011006. [CrossRef]
5. Hodas, S.; Izvoltova, J.; Rekus, D. Trends in Inertial Navigation Technologies. *Earth Environ. Sci.* **2021**, *906*, 012069. [CrossRef]
6. Al-Radaideh, A.; Al-Jarrah, M.; Jhemi, A. UAV testbed building and development for research purposes at the american universityof sharjah. In Proceedings of the ISMA'10 7th International Symposium on Mechatronics and Its Applications, Sharjah, United Arab Emirates, 20–22 April 2010.
7. Al-Radaideh, A. Guidance, Control and Trajectory Tracking of Small Fixed Wing Unmanned Aerial Vehicles (UAV's). Master's Thesis, American University of Sharjah, Sharjah, United Arab Emirates, 2009.
8. Laupré, G.; Khaghani, M.; Skaloud, J. Sensitivity to time delays in VDM-based navigation. *Drones* **2019**, *3*, 11. [CrossRef]
9. Bassolillo, S.R.; D'Amato, E.; Notaro, I. Enhanced Attitude and Altitude Estimation for Indoor Autonomous UAVs. *Drones* **2022**, *6*, 18. [CrossRef]
10. Papafotis, K.; Sotiriadis, P.P. Exploring the Importance of Sensors' Calibration in Inertial Navigation Systems. In Proceedings of the 2020 IEEE International Symposium on Circuits and Systems (ISCAS), Seville, Spain, 12–14 October 2020.
11. Grewal, M.S.; Andrews, A.P.; Bartone, C.G. *Global Navigation Satellite Systems, Inertial Navigation, and Integration*, 3rd ed.; John Wiley & Sons: New York, NY, USA, 2013; pp. 54–103.
12. Groves, P.D. Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems. *IEEE Aerosp. Electron. Syst. Mag.* **2015**, *30*, 26–27. [CrossRef]
13. Wang, X.J.; Li, D.; Zhang, X.; Sheng, D.Y.; Li, Z. Research and experiment on system level calibration method of fiber optic strapdown inertial navigation system. In Proceedings of the 2019 2nd International Conference on Information Systems and Computer Aided Education (ICISCAE), Dalian, China, 28–30 September 2019.
14. Han, S.; Meng, Z.; Omisore, O. Random Error Reduction Algorithms for MEMS Inertial Sensor Accuracy Improvement—A Review. *Micromachines* **2020**, *11*, 1021. [CrossRef]
15. Wang, Y.; Lin, Y.W.; Askari, S.; Jao, C.S.; Shkel, A.M. Compensation of systematic errors in ZUPT-aided pedestrian inertial navigation. In Proceedings of the 2020 IEEE/ION Position, Location and Navigation Symposium (PLANS), Portland, OR, USA, 20–23 April 2020.
16. Xiao, Y.; Ruan, X.; Chai, J.; Zhang, X.; Zhu, X. Online IMU self-calibration for visual-inertial systems. *Sensors* **2019**, *19*, 1624. [CrossRef]
17. Abro, G.E.M.; Zulkifli, S.A.B.; Asirvadam, V.S. Dual-loop single dimension fuzzy-based sliding mode control design for robust tracking of an underactuated quadrotor craft. *Asian J. Control* **2022**, 1–26. [CrossRef]
18. Abro, G.E.M.; Zulkifli, S.A.B.; Asirvadam, V.S.; Ali, Z.A. Model-free-based single-dimension fuzzy SMC design for underactuated quadrotor UAV. *Actuators* **2021**, *10*, 191. [CrossRef]
19. Chen, C.; Zhao, P.; Lu, C.X. Deep-learning-based pedestrian inertial navigation: Methods, data set, and on-device inference. *IEEE Internet Things J.* **2020**, *7*, 4431–4441. [CrossRef]
20. Wang, J.; Lou, W.; Liu, W. Calibration of MEMS Based Inertial Measurement Unit Using Long Short-Term Memory Network. In Proceedings of the 2019 IEEE 14th International Conference on Nano, Bangkok, Thailand, 11–14 April 2019.
21. Weber, D.; Gühmann, C.; Seel, T. Neural networks versus conventional filters for Inertial-Sensor-based attitude estimation. In Proceedings of the 2020 IEEE 23rd International Conference on Information Fusion (FUSION), Rustenburg, South Africa, 6–9 July 2020.
22. Li, Q.; Ma, Z. A Hybrid Dynamic Probability Mutation Particle Swarm Optimization for Engineering Structure Design. *Mob. Inf. Syst.* **2021**, *2021*, 6648650. [CrossRef]

23. Zhang, H.; Yuan, M.; Liang, Y. A novel particle swarm optimization based on prey–predator relationship. *Appl. Soft Comput.* **2018**, *68*, 202–218. [CrossRef]
24. Zaman, H.R.; Gharehchopogh, F.S. An improved particle swarm optimization with backtracking search optimization algorithm for solving continuous optimization problems. *Eng. Comput.* **2021**, 1–35. [CrossRef]
25. Batista, P.; Silvestre, C.; Oliveira, P.; Cardeira, B. Accelerometer calibration and dynamic bias and gravity estimation: Analysis, design, and experimental evaluation. *IEEE Trans. Control Syst. Technol.* **2010**, *19*, 1128–1137. [CrossRef]
26. Beravs, T.; Podobnik, J.; Munih, M. Three-axial accelerometer calibration using Kalman filter covariance matrix for online estimation of optimal sensor orientation. *IEEE Trans. Instrum. Meas.* **2012**, *61*, 2501–2511. [CrossRef]
27. Xiang, S.; Xing, L.N.; Wang, L.; Zhou, K. Comprehensive learning pigeon-inspired optimization with tabu list. *Sci. China Inf. Sci.* **2019**, *62*, 07028. [CrossRef]
28. Li, C.; Duan, H.B. Target detection approach for UAVs via improved pigeon-inspired optimization and edge potential function. *Aerosp. Sci. Technol.* **2014**, *39*, 352–360. [CrossRef]
29. Zhan, Z.H.; Zhang, J.; Li, Y.; Chung, H.S. Adaptive particle swarm optimization. *IEEE Trans. Syst. Man Cybern. Part B* **2009**, *39*, 1362–1381. [CrossRef] [PubMed]
30. Tian, D.; Shi, Z. MPSO: Modified particle swarm optimization and its applications. *Swarm Evol. Comput.* **2018**, *41*, 49–68. [CrossRef]
31. Duan, H.B.; Qiao, P. Pigeon-inspired optimization: A newswarm intelligence optimizer for air robot path planning. *Int. J. Intell. Comput. Cybern.* **2014**, *7*, 24–37. [CrossRef]
32. Duan, H.B.; Qiu, H.X.; Fan, Y.M. Unmanned aerial vehicle formation cooperative control based on predatory escaping pigeon-inspired optimization. *Sci. Sin. Technol.* **2015**, *45*, 559–572.
33. Yuan, G.S.; Xia, J.; Duan, H.B. A continuous modeling method via improved pigeon-inspired optimization for wake vortices in UAVs close formation flight. *Aerosp. Sci. Technol.* **2022**, *120*, 107259. [CrossRef]
34. Yang, Z.Y.; Duan, H.B.; Fan, Y.M. Automatic carrier landing system multilayer parameter design based oncauchy mutation pigeon-inspired optimization. *Aerosp. Technol.* **2018**, *79*, 518–530. [CrossRef]
35. Herdianti, W.; Gunawan, A.A.; Komsiyah, S. Distribution cost optimization using pigeon inspired optimization method with reverse learning mechanism. *Procedia Comput. Sci.* **2021**, *179*, 920–929. [CrossRef]